1989年2月1日発行(毎月1回1日発行)第8巻2号通巻82号　昭和58年11月2日第三種郵便物認可

ISSN 0910-7614

パーソナルコンピュータ・マガジン
MZシリーズ,X1/turbo,X68000&ポケコン

Oh!X

オー!エックス　定価540円



特集
マシン語“でじたるざんまい”

超入門Z80マシン語活用術
アセンブラによるX68000料理教室

S-OS全機種共通システム
エディタアセンブラREDA
X1版S-OS“SWORD”再掲載

X1/turboロールプレイングゲーム
FLAME

THE SOFTOUCH
原宿アフターダーク/Murder Club DX
極道陣取り/ザ・キングオブシカゴ
彩CRONE/Final Ver.3.2

LIVE in '89
X1/turboソーサリアン オープニングテーマ
X68000ニンジャウォーリアーズよりDuddy Mulk
MZ-2500ファンタジーゾーンよりHot Snow

猫とコンピュータ/知能機械概論
OS-9/X68000入門/C調言語講座PRO-68K
Z80マシン語ゲーム工房

2
FEB. 1989

SHARP New-Life People

20MBハードディスクモデル



X68000
PERSONAL WORKSTATION
ACE HD

通商産業省選定

■本体＋キーボード＋マウス・トラックボール
CZ-611C-GY(グレー)・-BK(ブラック)標準価格399,800円
写真はCZ-611C-GY＋CZ-601D-GY＋CZ-6ST1-E

ハイコストパフォーマンスFDモデル



X68000
PERSONAL WORKSTATION
ACE

■本体＋キーボード＋マウス・トラックボール
CZ-601C-GY(グレー)・-BK(ブラック)標準価格319,800円
写真はCZ-601C-BK＋CZ-603D-BK

〈X68000ACEシリーズの主な特長〉●実装密度をさらに追求して信頼性を高めたマンハッタンシェイプ ●68000搭載 ●テキスト、グラフィック、スプライト、独立3画面設計、最大12Mバイトの大容量メモリ(標準1Mバイト) ●フレンドリーOS、Human 68k搭載 ●連文節変換、マルチフォントをサポートした強力日本語処理 ●1024×1024ドット(最大表示エリア768×512ドット)の実画面エリアを装備した高解像度表示能力 ●512×512ドット、65,536色同時発色 ●水平32、1画面128のパワフルなスプライト機能 ●オーバースキャン機能を採用した512×512ドットレベルのスーパーインポーズ ●テキストビットマップ方式採用 ●8重和音ステレオFM音源搭載 ●音声デジタイズ記録AD PCM※採用 ●マウス・トラックボール標準装備 ●5インチ1MバイトFDD2基搭載 ●「X-BASIC」、「辞書ディスク」と各種ユーティリティ、「日本語ワードプロセッサ」をバンドル ●20Mバイトハードディスク内蔵(CZ-611C)

※Adaptive Differential PCM

■15型カラーディスプレイテレビ(ドットピッチ0.39mm)CZ-601D-GY(グレー)・-BK(ブラック)標準価格119,800円
■15型カラーディスプレイテレビ(ドットピッチ0.31mm)CZ-611D-GY(グレー)・-BK(ブラック)標準価格145,000円
■14型カラーディスプレイ(ドットピッチ0.31mm) CZ-603D-GY(グレー)・-BK(ブラック) 標準価格84,800円(チルトスタンド同梱)
■チルトスタンドCZ-6ST1-E(グレー)・-B(ブラック)標準価格5,800円(CZ-601D/611D用)

資料請求券
X68000
oh!X
2係

アートの領域へ。

クォリティを維持しつづけることは、ある意味では創造することより困難なこととも言われています。出会いが印象的であればあるほど、その後が大変です。このことは、そのままX68000の歩みを言い得ているかも知れません。確かに技術は日進月歩です。しかしそれだけでコンピュータがもつべき創造性を論ずることはできないのも、また事実です。私たちはテクノロジーとクリエイティブマインド、いわば人とマシンとのソフトウェアインターフェイスで応えます。ホリゾンタルなマシンとしての熟成。そこからはいくつもの分野が見えてくるはずです。そしてどんな分野にしろX68000の仕事はアートであるべきです──。ますます洗練されて信頼性を高めたACEシリーズの登場で、あなたはまた新たな可能性に出会えそうです。

豊富な周辺機器がクリエイティブワークをサポート。

●21型カラーディスプレイ	CU-21CD	標準価格139,800円
●RGBシステムチューナー	CZ-6TU	標準価格 35,800円
●カラーイメージスキャナ※1	CZ-8NS1	標準価格188,000円
●カラーイメージユニット※2	CZ-6VT1	標準価格 69,800円
●カラービデオプリンタ	CZ-6PV1	標準価格198,000円
●24ピン漢字プリンタ(80桁)	CZ-8PK7	標準価格122,000円
●24ピン漢字プリンタ(136桁)	CZ-8PK8	標準価格152,000円
●24ピン漢字プリンタ(80桁)	CZ-8PK9	標準価格 89,800円
●熱転写カラー漢字プリンタ	CZ-8PC3	標準価格 65,800円
●熱転写カラー漢字プリンタ	CZ-8PC2	標準価格 69,800円
●ハードディスクユニット(20MB)	CZ-620H	標準価格178,000円
●モデムユニット※3	CZ-8TM2	標準価格 49,800円
●RS-232Cケーブル(平行接続型)	CZ-8LM1	標準価格 7,200円
●RS-232Cケーブル(クロス接続型)	CZ-8LM2	標準価格 7,200円
●拡張I/Oボックス(4スロット)	CZ-6EB1	標準価格 88,000円
●1MB増設RAMボード(内蔵用)	CZ-6BE1A	標準価格 38,000円
●2MB増設RAMボード※4	CZ-6BE2	標準価格 79,800円
●4MB増設RAMボード※4	CZ-6BE4	標準価格138,000円
●FAXボード	CZ-6BC1	標準価格 79,800円
●MIDIボード	CZ-6BM1	標準価格 26,800円
●GP-IBボード	CZ-6BG1	標準価格 59,800円
●ユニバーサルI/Oボード	CZ-6BU1	標準価格 39,800円
●増設用RS-232Cボード(2チャンネル)	CZ-6BF1	標準価格 49,800円
●数値演算プロセッサボード	CZ-6BP1	標準価格 79,800円
●スキャナ用パラレルボード	CZ-6BN1	標準価格 29,800円
●システムラック	CZ-6SD1	標準価格 44,800円
●アンプ内蔵スピーカーシステム(2本1組)	AN-160SP	標準価格 59,800円
●マウス	CZ-8NM2A	標準価格 6,800円
●トラックボール	CZ-8NT1	標準価格 13,800円
●ジョイカード	CZ-8NJ1	標準価格 1,700円
●高性能CRTフィルター	BF-68PRO	標準価格 19,800円

※1 使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ転送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1で接続してください。※2 使用に際してはコンピュータ本体と専用15型カラーディスプレイテレビ(CZ-601D、CZ-611Dなど)が必要です。※3 モデムユニットCZ-8TM2に同梱のソフトはX1/X1 turboシリーズ用です。※4 使用に際しては、あらかじめ、別売の1MB増設RAMボードCZ-6BE1Aを増設してください。

アートツールと呼びたい「PRO-68K」シリーズソフト。

イージーオペレーションの統合型表計算ソフト BUSINESS PRO-68K	CZ-212BS	標準価格 68,000円
コマンド型リレーショナルデータベース DATA PRO-68K	CZ-220BS	標準価格 58,000円
ワープロ機能を備えたカード型リレーショナルデータベース CARD PRO-68K	CZ-226BS	標準価格 29,800円
FM音源をフルサポートするサウンドエディタ SOUND PRO-68K	CZ-214MS	標準価格 15,800円
マウスを使った簡単操作の楽譜ワープロ MUSIC PRO-68K	CZ-213MS	標準価格 18,800円
AD PCM機能をサポートしたサンプリングエディタ Sampling PRO-68K	CZ-215MS	標準価格 17,800円
オリジナリティを活かせるポップアートツール NEW Print Shop PRO-68K	CZ-221HS	標準価格 19,800円
スクリーンエディタ内蔵の通信ソフト Communication PRO-68K	CZ-223CS	標準価格 19,800円
ソフトウェア開発に役立つCコンパイラ C compiler PRO-68K	CZ-211LS	標準価格 39,800円
24トラックのMIDIマルチレコーディングソフト Musicstudio PRO-68K	CZ-237MS	標準価格 25,800円
MIDI楽器演奏が楽しめるMUSIC PRO68KのMIDI版 MUSIC PRO-68K〔MIDI〕	CZ-247MS	1月発売予定
ソフトウェア開発ツール THE 福袋 V2.0	CZ-224LS	標準価格 9,980円
マルチタスク、リアルタイムオペレーティングシステム OS-9/X68000	CZ-219SS	標準価格 29,800円
本格財務会計ソフトウェア TOP財務会計	CZ-227BS	標準価格200,000円
AI開発ツール AI-68K(Staff LISP/OPS PRO-68K)	CZ-234LS	標準価格188,000円

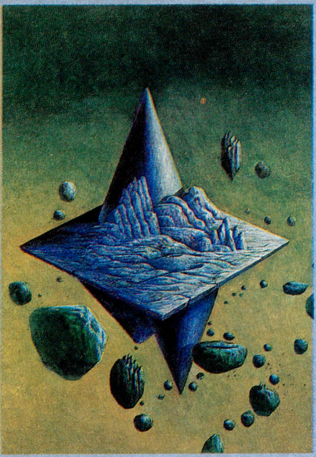
ゲームソフト

●ツインビー	CZ-217AS	標準価格 7,800円
●アルカノイド	CZ-222AS	標準価格 7,800円
●沙羅曼蛇	CZ-218AS	標準価格 8,800円
●熱血高校ドッジボール部	CZ-232AS	標準価格 7,800円
●フルスロットル	CZ-231AS	標準価格 8,800円

〈パソコン教室開催のお知らせ〉 X68000、MZ-2861のパソコン教室を開催します。くわしくは、下記までお問い合せください。

札幌(011)642-8111・仙台(022)288-8705・東京(03)260-1161・横浜(045)201-6525・名古屋(052)332-2611・大阪(06)222-7655・神戸(078)291-8715・福岡(092)481-2860

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部テレビ事業部第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)



表紙絵：Matsubaguchi Tadao

■広告目次

Oh!X

C O N T

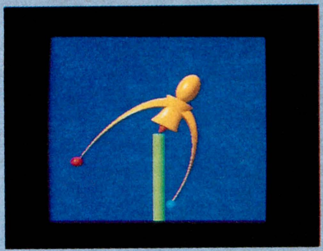
●特集

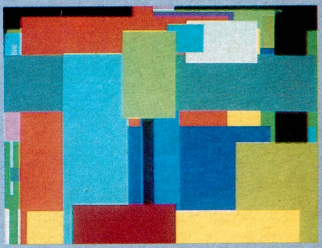
〈スタッフ〉
●編集長／前田 徹 ●副編集長／永野 仁 ●編集／植木章夫 石塚康世 高野庸一 ●協力／有田隆也 中森 章 清水和人 後藤貴行 林 一樹 浅野恵造 山村 一 井本 泰 堀内保秀 荻窪 圭 藤原和典 岡本浩一郎 毛内俊行 野中俊一郎 吉田賢司 影山裕昭 相馬英智 古村 聡 村田敏幸 倉持亮一 ●カメラ／杉山和美 ●イラスト／永沢しげる 山田晴久 小栗由香 ●アートディレクター／島村勝頼 ●レイアウト／元木昌子 AD GREEN ●校正／手塚喜美子 千野延明

1989 FEB.

2

E N T S

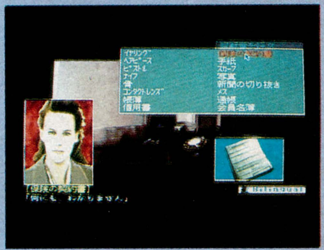

彩CRONE


特集 マシン語"でじたるざんまい"


Final Ver.3.2

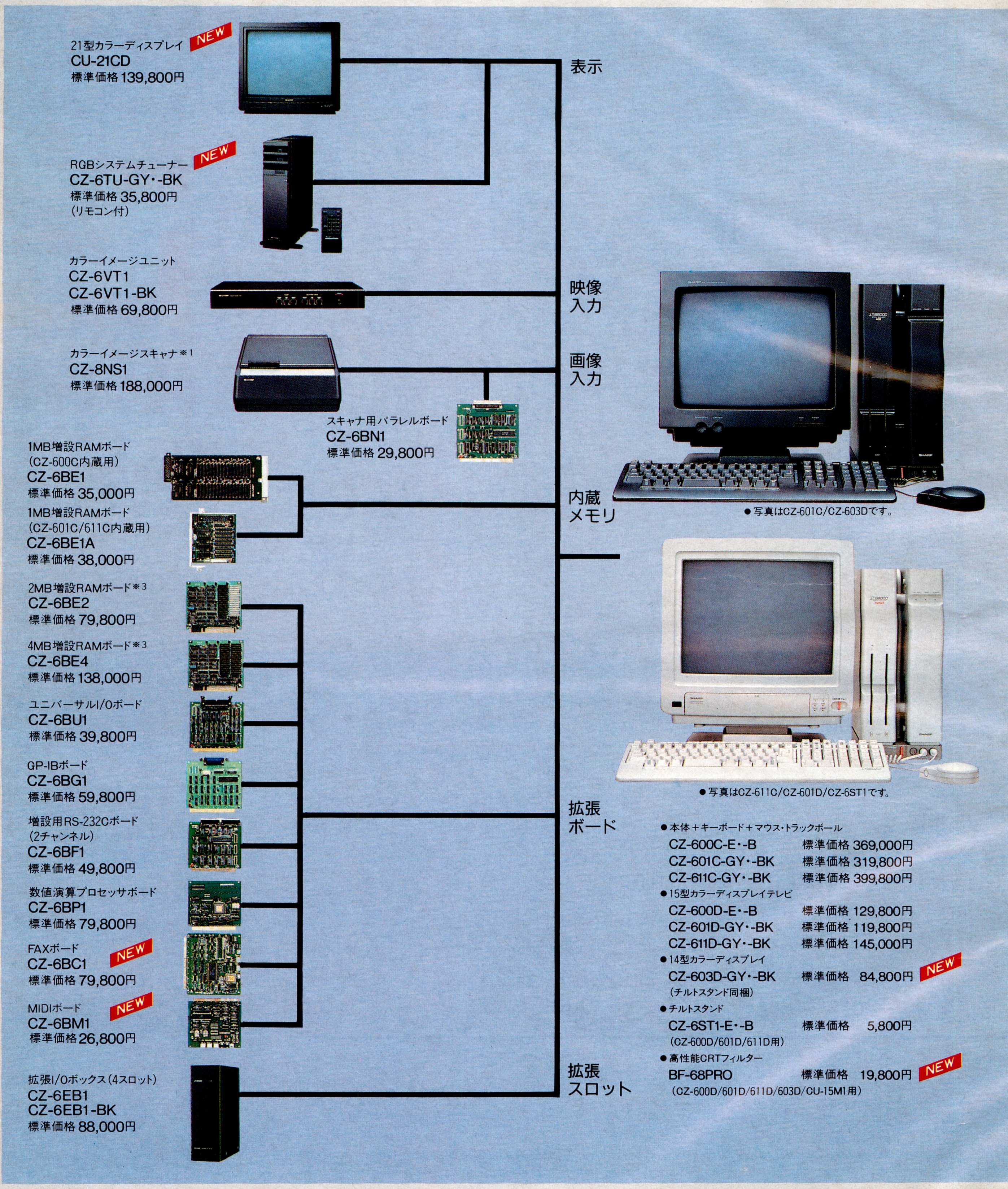

FLAME


原宿アフターダーク


Murder Club DX

SHARP

クリエイティブマインド



※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ伝送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1' 標準価格29,800円で接続してください。
CZ-6BE1標準価格35,000円(CZ-600C)、CZ-6BE1A 標準価格38,000円(CZ-601C、CZ-611C)を増設してください。

シャープ株式会社 ●お問い合わせは…… シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

資料請求券

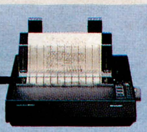
思わず熱くなる。

あふれる周辺機器がX68000をサポート。

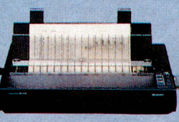


シャープペリフェラルファミリー X68000

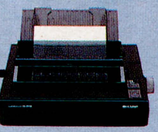
プリントアウト



24ピン漢字プリンタ(80桁)
CZ-8PK7
標準価格122,000円
(信号ケーブル同梱)



24ピン漢字プリンタ(136桁)
CZ-8PK8
標準価格152,000円
(信号ケーブル同梱)



24ピン漢字プリンタ(80桁)
CZ-8PK9
標準価格89,800円
(信号ケーブル同梱)



熱転写カラー漢字プリンタ
CZ-8PC2
標準価格69,800円
(信号ケーブル同梱)



NEW
熱転写カラー漢字プリンタ
CZ-8PC3
標準価格65,800円
(信号ケーブル同梱)

ビデオプリント



カラービデオプリンタ
CZ-6PV1
標準価格198,000円
(信号ケーブル同梱)

ハードディスク



ハードディスクユニット(20MB)
CZ-620H
標準価格178,000円

通信



RS-232Cケーブル
(平行接続型)
CZ-8LM1
標準価格7,200円

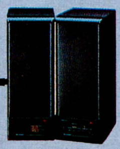


モデムユニット※2
CZ-8TM2
標準価格49,800円
(RS-232Cケーブル同梱)



RS-232Cケーブル
(クロス接続型)
CZ-8LM2
標準価格7,200円

サウンド



アンプ内蔵スピーカーシステム
(2本1組)
AN-160SP
標準価格59,800円

システムラック

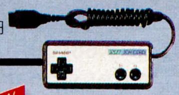


システムラック
CZ-6SD1
標準価格44,800円

入力



NEW
マウス
CZ-8NM2A
標準価格6,800円



ジョイカード
CZ-8NJ1
標準価格1,700円



NEW
トラックボール
CZ-8NT1
標準価格13,800円

※2 モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。　※3 ご使用に際しては、あらかじめ別売の1MB増設RAMボード

X1・X1turboシリーズ用周辺機器

カラーディスプレイ		
●21型カラーディスプレイ※1	CU-21CD	139,800円
映像・画像入力編集装置		
●カラー イメージスキャナ	CZ-8NS1	188,000円
●カラー イメージボードII	CZ-8BV2	39,800円
●立体映像セット	CZ-8BR1	29,800円
●パーソナルテロッパ※2	CZ-8DT2	44,800円
FM音源		
●ステレオタイプFM音源ボード	CZ-8BS1	23,800円
スピーカー(2本1組)標準装備、ミュージックツール同梱		
プリンタ		
●24ピン漢字プリンタ(80桁)	CZ-8PK7	122,000円
●24ピン漢字プリンタ(136桁)	CZ-8PK8	152,000円
●24ピン漢字プリンタ(80桁)	CZ-8PK9	89,800円
●熱転写カラー漢字プリンタ	CZ-8PC3	65,800円
●熱転写カラー漢字プリンタ	CZ-8PC2	69,800円
●カラービデオプリンタ	CZ-6PV1	198,000円
ファイル		
●ミニフロッピーディスクユニット(2HD・2D)※3	CZ-520F	118,000円
●ミニフロッピーディスクユニット(2D)	CZ-502F	99,800円
●ミニフロッピーディスクユニット(2D・1ドライブ)	CZ-503F	49,800円
●増設用ミニフロッピーディスクドライブ(2D)※4	CZ-53F-BK	19,800円
●ハードディスクユニット※3	CZ-500H	348,000円
●ハードディスクユニット(増設用)	CZ-501H	258,000円
●カセットデータレコーダ	CZ-8RL1	24,800円
●ミニフロッピーディスク	CZ-5M2D/CZ-5M2HD(各10枚入)	
●コンパクトフロッピーディスク	CZ-3FBD	1,300円
拡張ボード・その他		
●モデムユニット(300/1200ボー)	CZ-8TM2	49,800円
●320KB外部メモリ	CZ-8BE2	29,800円
●ROM BASICボード※5	CZ-8RB	19,800円
●RS-232C・マウスボード※6	CZ-8BM2	19,800円
●フロッピーディスクインターフェイス※7	CZ-8BF1	14,800円
●JIS第1水準漢字ROM※8	CZ-8BK2	19,800円
●JIS第2水準漢字ROM※9	CZ-8BK4	6,800円
●JIS第2水準漢字ROM&ターボ博士レキシコン・日本語百科ワードパワー※10	CZ-8BK3	13,800円
●RS-232C用ケーブル(平行接続型)	CZ-8LM1	7,200円
●RS-232C用ケーブル(クロス接続型)	CZ-8LM2	7,200円
●拡張I/Oボックス	CZ-8EB3	33,800円
●RFコンバータ※11	AN-58C	2,980円
●マウス	CZ-8NM2A	6,800円
●トラックボール	CZ-8NT1	13,800円
●ジョイカード	CZ-8NJ1	1,700円
●チルトスタンド※12	CZ-6ST1-E･-B	5,800円
●高性能CRTフィルター	BF-68PRO	19,800円
●システムスタンド	CZ-8SS2	5,500円
●スキャナ用パラレルボード※13	CZ-8BN1	27,800円

(価格は標準価格です。)

●品番中の－表示は、B〈ブラック〉・E〈オフィスグレー〉を示します。※1 X1ターボZシリーズ用 ※2 CZ-862Cには接続できません ※3 X1ターボシリーズ用 ※4 CZ-830C用 ※5 X1シリーズ用V1.0 ※6 X1シリーズ用 ※7 CZ-850CでCZ-520Fを使用する場合に必要 ※8 CZ-800C、801C、802C、803C、811C、820C用 ※9 CZ-856C用 ※10 CZ-850C、851C、852C、862C用 ※11 CZ-820C、822C、830C用 ※12 CZ-600D、601D、611D、880D、830D、CU-15M1用 ※13 CZ-8NS1用 ●接続等の説明につきましては、周辺機器総合カタログをご参照ください。

電子機器事業本部テレビ事業部第4商品企画部〒162東京都新宿区市谷八幡町8番地☎(03)260-1161(大代表)

SHARP

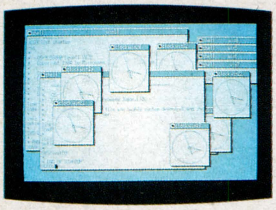
"アート"と呼べる高水準のソフトウェアが

OS-9/X68000

EWSレベルのプロフェッショナルな

パーソナルウィンドウ

オーバーラッピングタイプのマルチウィンドウ「パーソナルウィンドウ」をサポート。メモリの許す限りいくつでも開くことができます。これらのウィンドウは独立したディスプレイ感覚で自由に利用でき、プログラミングレベルからアプリケーションレベルまで使いやすく機能的な環境を提供します。ウィンドウまわりのユーザーインターフェイスは、ポップアップメニュー中心に構成され、マウスカーソルの位置によって、実行中のタスクを自由に切り換えたり、不必要なウィンドウをアイコン化して必要なときに開いたりすることもできます。また表示環境には"アクション・リージョン"という概念を導入、表示画面上のマウス・カーソル位置に特定の意味を持たせて、他のプログラムを起動したり、シグナルを送って同期をとることも可能です。



マルチタスク/マルチウィンドウ、ハイパーメディア環境をサポート、X68000のハイアビリティに応えるオペレーティングシステム。

グラフィックもサウンドも

マルチメディア対応

X68000のもつ優れたグラフィック機能はもちろん、AD PCMやFM音源もOSレベルでサポート。高度でしかもフレンドリーなハイパーメディア環境を実現しています。グラフィックとサウンドのマルチタスク処理も可能、タイミングの同期も簡単にはかれます。またFM音源に対しては、Human 68k上の「MUSIC PRO-68K」など、他のソフトウェアで作成されたファイルをメディアコンバートして実行できます。

ファイルの互換性 標準でHuman68k、MS-DOSファイルのリード/ライトがサポートされていますので、すでにお持ちのデータを容易にコンバートできます。

日本語処理機能 Human68kの日本語処理であるASK68Kと同等のものが組込まれていますので、使い勝手が同じであるばかりでなく、Human68kで使い込んだ辞書ファイルをコンバートすることにより、OS-9で利用することができます。

マルチタスク・リアルタイムI/O すべてのI/Oデバイスがマルチタスク、リアルタイム機能でサポートされています。

AV-RIDER OS-9のAV機能をフル活用したサンプルソフト。ファイル起動の機能を持ったエンターテイメントソフトウェアで、オーディオファイル、ビデオファイル及びオーディオ・ビデオ混在ファイルをセレクト実行します。

- OS-9はマイクロウェア社の登録商標です。
- MS-DOSは米国マイクロソフト社の登録商標です。



マルチタスク、リアルタイムオペレーティングシステム

OS-9/X68000 CZ-219SS 標準価格 29,800円

OS-9/X68000用開発ツール
● C & Professional Pack 58,000円
〔発売元:マイクロウェアジャパン㈱〕

資料請求券
CZソフト
oh!X
2係

X68000をサポート。



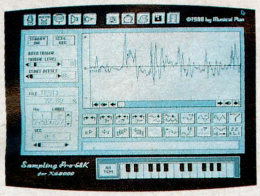
シャープオリジナルソフトウェア X68000

サウンドツール

Sampling PRO-68K

AD PCM機能をフルに活かす高機能サンプリングエディタ。録音した音声を波形表示し、それをエディットできるWAVE EDITOR、50音データでX68000がしゃべるSPEACH EDITORなどをサポート。サンプリング音のデータはBASICでも活用できます。

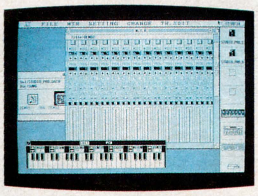
■CZ-215MS 標準価格 17,800円



Musicstudio PRO-68K

24の録音トラックをもったプロフェッショナルユースのMIDIマルチレコーディングソフトウェア。MIDI楽器を使って演奏したデータをスタジオ感覚で編集、記録・再生できます。キーボードによるリアルタイムレコードをサポートする多彩な機能を装備しています。

■CZ-237MS 標準価格 25,800円



MUSIC PRO-68K〔MIDI〕

MUSIC PRO-68KのMIDI版。MIDI対応自動伴奏機能をサポート、簡単な楽譜入力でMIDI楽器演奏が楽しめます。

■CZ-247MS 1月発売予定

MUSIC PRO-68K

最大8パートのスコア(総譜)が書け、内蔵のFM音源で演奏できる楽譜ワープロ&演奏用ツール。

■CZ-213MS 標準価格 18,800円

SOUND PRO-68K

スタジオのコンソールパネルを操作する感覚でFM音源による音づくりが楽しめるサウンドエディタ。

■CZ-214MS 標準価格 15,800円

アートツール

NEW Print Shop PRO-68K

オリジナリティあふれるはがき等、簡単に作成、印刷できるホームプロダクティビティツール。ほとんどの処理をアイコンで表示しマウスで選ぶフレンドリーなオペレーション。

■CZ-221HS 標準価格 19,800円



グラフィックライブラリ VOL.1

暑中見舞用を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。

■CZ-235GS 標準価格 8,800円

グラフィックライブラリ VOL.2

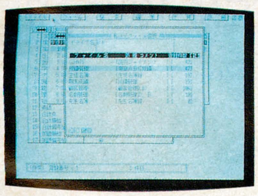
クリスマス、年賀状用を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。■CZ-236GS標準価格8,800円

ビジネスツール

CARD PRO-68K

プルダウン、アイコン、ポップアップなど多彩なメニュー表示によるヒューマンな操作性を実現した高性能カード型リレーショナルデータベース。ウィンドウ間のデータのやりとりも自由にできます。顧客管理、住所録管理、システム手帳への応用など幅広く活用できます。

■CZ-226BS 標準価格 29,800円



DATA PRO-68K

コマンド入力の手間を軽減するヒストリー機能、罫線ドライバー付レポートライター機能、10進31桁の高度な演算精度。さらにイメージ表示機能を装備したコマンド型リレーショナルデータベースです。

■CZ-220BS 標準価格 58,000円

BUSINESS PRO-68K

スプレッドシート(表計算)、データベース、グラフ作成機能を緊密に一体化させた統合ビジネスツールです。マウス対応のやさしいオペレーション、高度なエディタ機能、豊富な関数群も装備しています。

■CZ-212BS 標準価格 68,000円

TOP財務会計

会計エキスパートシステムとデータベースを搭載し、機能と操作性の向上を両立させた財務会計ソフトウェアです。

■CZ-227BS 標準価格200,000円

システム手帳リフィル集

スケジュール管理、名刺管理、金銭管理など総合的にマネジメントできるCARD PRO-68K用リフィル集。

■1月発売予定

データフォーム集

給与管理、売上管理、見積書などのデータフォームを集めたCARD PRO-68K用ツールです。

■1月発売予定

開発ツール

C compiler PRO-68K

Cコンパイラ(XC)やBASIC-Cコンバータ(XBAStoC)などからなる開発ツール。Human 68k上におけるプログラム開発を効率良くサポートします。XCは、C言語の最も基本的な仕様(K&R)に準拠し、ANSI仕様も採り入れた最新バージョン。豊富なライブラリ(約700種)も用意されています。

■CZ-211LS 標準価格 39,800円

THE 福袋 V2.0

アセンブラ、リンカ、デバッガ、アーカイバ、X-BASIC V2.00からなる開発ツール。

■CZ-224LS 標準価格 9,980円

AI-68K (Staff LISP/OPS PRO-68K)

AI開発用言語とエキスパート構築ツールがセットになったAIプログラム開発ツールです。

■CZ-234LS 標準価格188,000円

通信ツール Communication PRO-68K

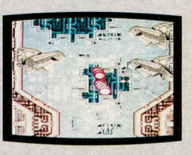
300～19,200BPSまでの通信速度に対応し、各種データベースの漢字端末やパソコン通信に利用できます。逆スクロール機能、自動実行機能、コンカレント機能も装備。さらに行入力機能やスクリーンエディタなど豊富な編集機能をもった高機能通信ソフト。

■CZ-223CS 標準価格 19,800円

ゲーム



ツインビー
シューティングゲーム
■CZ-217AS
標準価格7,800円



沙羅曼蛇
シューティングゲーム
■CZ-218AS
標準価格8,800円



アルカノイド
ブロックゲーム
■CZ-222AS
標準価格7,800円



フルスロットル
ドライブゲーム
■CZ-231AS
標準価格8,800円



熱血高校ドッジボール部
スポーツゲーム
■CZ-232AS
標準価格7,800円

各ソフトハウスのアプリケーションも続々登場

●日本語ワープロ		
EW(イー・ダブリュー)	38,000円	イースト
●表集計・データベース		
ビジレスAD68K	98,000円	MASH SYSTEMS
Kamikaze(神風)	68,000円	サムシンググッド
●グラフィックツール		
C-TRACE 68	68,000円	キャスト
Z'S STAFF PRO-68K	58,000円	ツァイト
彩crone 68K	58,000円	アンスコンサルタンツ
G 68K	14,800円	OH!BUSINESS
●通信ソフト		
X Talk-68K	12,800円	SSKコンピュータ
PCOM68αI	55,000円	パーソナルビジネスアシスト
●OS		
CONCERTO-X68K	99,800円	アクセス
●システムユーティリティ		
WINDEX PRO-68K	28,000円	ジェー・イー・エル
Toy & Tools	6,800円	計測技研
●プログラム言語		
CMA68K	29,800円	シティソフト
●教育		
がっちり君シリーズ英語	各5,000円	SEICHI
パル英単語シリーズ	各9,000円	パル教育システム
●ゲーム		
A列車で行こうII	12,800円	アートディンク
ザ・リターン・オブ・イシター	7,800円	エス・ピー・エス
億万長者	9,800円	コスモスコンピューター
ソフトでハードな物語	7,800円	システムサコム
プロフェッショナル麻雀悟空	7,800円	シャノアール
ドラゴンスピリット	8,800円	電波新聞社
スペースハリヤー	6,800円	電波新聞社
源平討魔伝	7,800円	電波新聞社
たんば	7,800円	マイクロネット
琥珀色の遺言	9,800円	リバーヒルソフト

※この他、既発売、発売予定のソフトが約330本。詳しくはX68000シリーズ用「SOFTWARE FIELD」VOL.15をご参照下さい。

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部テレビ事業部 第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)へ。



この画像はC-TRACEで作成したものです。(背景はスーパータブロー)

プロフェッショナル「CG作家」募集!!

★資格：C-TRACE登録ユーザーであること。

★目的：職業作家としての自立を援助し、CGを広く世間一般に普及させる。

★方法：C-TRACEにより作られたデータを当社のSYSTEM-100により高画質演算し、この作品をフォトライブラリーにて運用する。
また、依頼制作も受注する。

★費用：作家は一切を必要としません。
データを送り、運用益を受取るだけです。
演算料金、プリント費用、その他すべて当方で負担いたします。

●資料請求は、切手を貼り宛先を記入した返信用封筒を同封の上、下記までお送り下さい。

※C-TRACEユーザーは、シリアルNo.を明記して下さい。

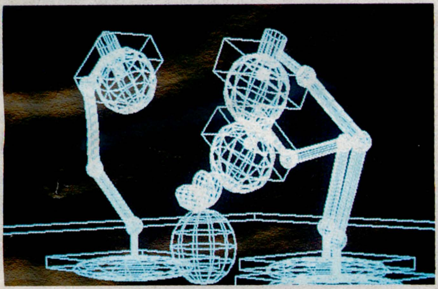
〒158 東京都世田谷区等々力2-1-13
キャスト「プロ作家募集」係

CasT
株式会社 キャスト



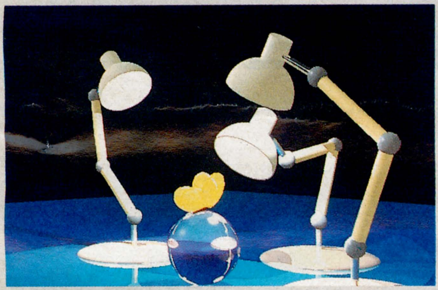
モデリング新時代！

レイトレースは数学的なロジックなので、それなりに高度なものになっていきます。しかし、モデリングの使い勝手と自由度がどこのメーカーでもうまくありません。
サイクロンはこの問題解決を優先してきました。どんなすばらしいイメージでも、モデリングの煩雑さで大きな制限を受けることはファイト(製作意欲)をなくすことであり、イメージをつまらせてしまいます。



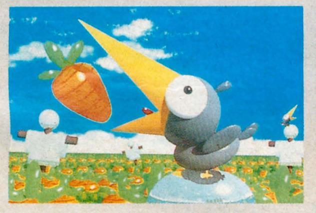
ここが違う、サイクロン!! —モデリングがすべてを決める！—

- このデータ**モデリング15分！**他では考えられない**簡単さ。**
- ANS独自の**階層構造型モデラー**で、リスト記述は**いっさい不要。**
- 4面ワイヤーフレームで**ビジュアル形状確認。**(3面図、パース図)
- **サイクロンだけ!!** 論理演算で(カッコ)が使える。



アニメーションも可能!!強力なマクロ編集機能

- 単体プリミティブの色・質感**まとめて変更。**
- 物体の**配置変更**自由自在。
- 結局部品が生きる**ライブラリー。**






対応機種

NEC：PC-98VM、VX、RX、RA
数値データプロセッサー付きに限る。
フレームバッファが必要です。

シャープ：X68000

サイクロン・ファミリー

※オープンアーキテクチャーを基本コンセプトにしています。

PSY-CRONE 68K(SHARP X68000)……………58,000円
- 彩croneモデラー
- 彩croneレンダラー

X68000で気軽に3次元CGを楽しみたい方へ。(Z's STAFF PRO68Kとの連動も可能です。)

PSY-CRONE LIGHT(PC-9801)……………………98,000円
- 彩croneモデラー
- 彩croneレンダラー

これから3次元CGに取り組もうと考えている方への入門版。(スーパータブローとの連動も可能です。)

PSY-CRONE TURBO(PC-9801)……………… 250,000円
- 彩croneモデラー
- 彩croneレンダラー
- REY-TREKⅡフィルター
- ユーティリティ

各種ユーティリティプログラムとRAY-TREKⅡ(VI社)との連動も可能なフィルターがついた中級版。

PSY-CRONE SUPER(PC-9801)…………… 1,500,000円
- 彩croneモデラー
- 彩croneスーパーレンダラー
- 彩croneスーパーペインティング
- フレームバッファ
- ユーティリティ

3次元CGソフトの最高峰。フルカラーCGの世界を極めたい方にセットで。

サイクロンデモンストレーションはお近くのNEC Bit-INNかSHARP OAルームまで。

サイクロン・フレンド

※RAY-TREKⅡ(PC-9801)(VI社)
※スーパータブロー(PC-9801)(SAPIENCE社)
Z's STAFF PRO68K(X68000)(zeit社)
AUTO CAD(PC-9801)(AutoDesk社)予定
サイクロンアニメーション(開発中)。
その他各CG商品とのインターフェース開発中。

※印はアンス・コンサルタンツの取扱い商品です。
製品改良のため仕様の一部を予告なく変更することがあります。

株式会社アンス・コンサルタンツ
九州本社/〒810 福岡市中央区平丘町68
phone.092-522-6347 FAX092-521-0400

What!

OH BUSINESSから 秋の新作2本!…

▶Easy Graphic Tool

あなたのイメージをかたちにするのがグラフィックツール

超多機能でも、つかいこなせないツールたち…………………
機能は小さくてもいいのです。つかいやすければ …………
G68Kのいのちはし・な・や・か・さです。



発売中!

■G68K機能一覧

●にじみ表現が可能なペン●エアブラシ●直線を引く●長方形を塗りつぶす●拡大・縮少●左右反転●上下反転●複写●塗りつぶし●2つの色を混ぜ合わせ新しい色を作る● イメージスキャナ(GT-3000)をサポート●内山亜紀先生の緻密で綺麗なイラストデータ入り●作業中のBGM付きグラフィックツール (選曲可能)

●Z's STAFF PRO68Kのデータをロードセーブ
●アートマスター400(9801)からZ's STAFF PRO68Kへのデータコンバート機能
●アートマスター400(PC-9801用)のデータをロード

定価 ¥14,800

▶スプライトエディタ E68K

簡単にできる貴方だけのオリジナルグラディグス



発売中!

●65536色をサポート●
1つのスプライトに65536色中16色を選択して、1ドット単位で色が付けられます。
●1画面上で64パターンを同時編集●
1画面で64パターンのスプライトデータを編集できます。(1パターン 16*16ドット)
(ページ切り換え機能により28ページまでメモリーに保存できます)
●アニメーション機能を搭載●
作成したスプライトパターンを8コマまで設定し、動きを決めるとアニメーションできます。(作成したスプライトの動きがすぐに確認できます)
●拡大モードは4種類●
2・4・8・16倍で拡大エディットできます。
●強力な編集機能●
LINE・BOX・BOX FILL・PSETをサポートしています。
●BGM機能●
スプライトエディタでは初のBGM機能搭載。(5曲の中から選曲可能)
●スプライトデータならどんな形式でもエディット可能●
ディスク・メモリーからのスプライトデータの読み込みが可能です。
●増設RAM・ハードディスクをサポート●
増設RAMを接続していると1度にエディットできるデータ量が増えます。ハードディスクからの立ち上げ、ハードディスクからのデータ読み込みもOKです。
●開発にも最適●
E68K用市販ゲームのキャラクターを自由に変更して、貴方だけのオリジナルゲームを作る事も簡単にできてしまいます。

定価 ¥19,800

Ⓐ OH!ゲームコンテスト
・コンテスト入賞者には賞金が支払われます。
・詳しい応募要領をご希望の方は当社右の住所までお申し込み下さい。
(入賞作品の版権はSYSTEM HOUSE OHに帰属いたします)

貴方のアイデアをお待ちしております。
(PRG・AVG・ADULT・ジャンルを問いません)

I WANT YOU!

募集期間 '88.12月〜'89.4月末日〆切

Ⓑ OH!シナリオコンテスト

NEW SYSTEM HOUSE OH! (ニュー システム ハウス オー)
京都市山科区音羽西林町2 TEL:075-502-2972

販売代理店:近畿システムサービス㈱

京都市山科区音羽西林町2 TEL:075-502-2972 **OH! BUSINESS**

プロ指向のワークステーションをお届けします。

X68000上のソフト開発を強力にサポート

C&プロフェッショナル・パッケージ

C&プロフェッショナル・パッケージ

¥58,000

OS-9／X68000上で動作するマイクロウェア・Cコンパイラとユーティリティ・ソフトのパッケージです。動作させるためにはOS-9／X68000が必要です。

マイクロウェア・Cコンパイラは、K&RとANSIに準拠したスタンダード・Cライブラリ群とX68000のために拡張した豊富なライブラリ群とをサポートした強力なCコンパイラです。X68000のために拡張されたライブラリとしては、●フロッピー／ハードディスク●日本語処理　●ADPCM　●FM音源　●マウス　●ジョイスティック　●グラフィクス　●PSS（プレゼンテーション・サポート・システム）があります。

■ソフトウェア　●Cコンパイラ　●標準ライブラリ　●OS-9／X68000専用ライブラリ　PSS(プレゼンテーション・サポート・パッケージ)サポート・ライブラリ　●ヘッダファイル　●マクロ・アッセンブラR68　●リンカL68　●MAKEコマンド　●ユーザステイト・シンボリックデバッガ DEBUG　●漢字フル・スクリーンエディタKμMACS　●フル・スクリーンエディタSCRED

■マニュアル　●Cコンパイラ・ユーザーズ・マニュアル　●付属ユーティリティ・マニュアル　●OS-9／X68000専用ライブラリ・マニュアル

プログラマーズ・ツール・キット

OS-9／X68000のドライバ開発を強力に支援するパッケージです。

■ソフトウェア　●マクロアセンブラR68　●リンカL68　●MAKEコマンド　●漢字フル・スクリーンエディタKμMACS　●フル・スクリーンエディタSCRED　●システムステート・シンボリック・デバッガSYSDBG　●ROMデバッガROMDBG　●SCF系ドライバ・サンプル・ソースコード　●システム・デフィニション・ソース・コード

■マニュアル　●OS-9／X68000テクニカル・マニュアル　●システムインプリメンテーション・マニュアル　●付属ユーティリティ・マニュアル

OS-9/X68000　SRCDBG

OS-9／X68000用Cコンパイラに完全に準拠したソース・レベル・デバッガです。

C言語で作成したプログラムのデバッグが迅速に行なえます。ブレークポイントを設定しての実行はもちろん、変数をモニターしながらの実行、1ステップづつの実行、スタック・フレームの表示など20種類以上のコマンドが用意されています。

OS-9/X68000ネットワーク・システム

ローカル・エリア・ネットワーク（LAN）を構築するためのシステムです。

基本パッケージは、2枚のX68000用LANカードとケーブル、ソフトウェアのセットです。LANを構築することにより、1台のハードディスクやプリンタなどを複数台のX68000で共有できます。

OS-9 / X68000

microware®

マイクロウェア・ジャパン株式会社
〒273 千葉県船橋市本町4-41-19　Phone：0474-22-1747



7号店は、外から見て2FにX68000のシンボル「ツタンカーメン像」がめじるしです。

ツクモは決算! 大棚ざらえ

ツクモが丸ごと決算特価になった!!

40余年の感謝をこめて **同時開催** ツクモ創業記念感謝セール

2月4日(土)~12日(日)

期間中お買い上げの方に素敵なプレゼントをご用意してお待ちしています。

X68000 ACEシリーズ 好評発売中!

X68000 ACE HD CZ-611C (20MBハードディスク内蔵タイプ)……定価¥~~399,800~~

X68000 ACE CZ-601C (標準タイプ)……定価¥~~319,800~~

しっかりものディスプレイ他

CU-21CD 21型カラーディスプレイ……定価¥~~139,800~~
CZ-601D ドットピッチ0.39ミリ……定価¥~~119,800~~
CZ-611D ドットピッチ0.31ミリ……定価¥~~145,000~~
CZ-603D ドットピッチ0.31ミリ……定価¥~~84,800~~

■オプション

CZ-6ST1 チルト台……定価¥~~5,800~~
CZ-6TU RGBシステムチューナー……定価¥~~35,800~~
BF-68PRO 14・15インチCRTフィルター……定価¥~~19,800~~

豊富な周辺機器

CZ-6BE1 1MB増設RAMボード(CZ-600C用)…定価¥~~35,000~~
CZ-6BE1A 1MB増設RAMボード(ACEシリーズ用)……定価¥~~38,000~~
CZ-6BE2 2MB増設RAMボード……定価¥~~79,800~~
CZ-6BE4 4MB増設RAMボード……定価¥~~138,000~~
CZ-6BC1 FAXボード……定価¥~~79,800~~
CZ-6BP1 数値演算プロセッサボード……定価¥~~79,800~~
CZ-6VT1 カラーイメージユニット……定価¥~~69,800~~
CZ-8NS1 カラーイメージスキャナー……定価¥~~188,000~~
CZ-6BM1 MIDIボード……定価¥~~26,800~~

決算特価! おすすめソフトウェア

Kamikaze(神風) 統合型スプレッドシート……決算特価¥57,800
SOUND PRO-68K サウンドエディタ……定価¥~~15,800~~
MUSIC PRO-68K ミュージックツール……定価¥~~18,800~~
Sampling PRO-68K AD PCM活用ソフト……定価¥~~17,800~~
Musicstudio PRO-68K MIDIマルチレコーディングソフト…定価¥~~25,800~~
MUSIC PRO-68K [MIDI] MUSIC PRO-68KのMIDI版……近日発売
Communication PRO-68K 通信ソフト……定価¥~~19,800~~
DATA PRO-68K リレーショナルデータベース……定価¥~~58,000~~
CARD PRO-68K カード型データベース……定価¥~~29,800~~
Z's STAFF PRO-68K グラフィックツール……決算特価¥49,000
New Print Shop PRO-68K 高機能ポップアートツール…定価¥~~19,800~~
彩CRONE レイトレーシングソフトウェア……決算特価¥49,300
C COMPILER PRO-68K C言語開発セット……定価¥~~39,800~~
AI-68K AIプログラム開発ツール……定価¥~~188,000~~
OS-9/X68000……定価¥29,800

その他、ビジネスソフト・ホビーソフトも多数発売中ですので、お気軽にお訪ねください。



新登場! X1 turbo Z III

- CZ-888B-BK……¥169,800
- CZ-860D-BK……¥99,800

★上記セットに買い換えるなら **決算特価販売中**

下取り機種	差額
CZ-852C+CZ-850D	¥172,000
CZ-856C+CZ-870D	¥170,000
CZ-822C+CZ-820D	¥190,000

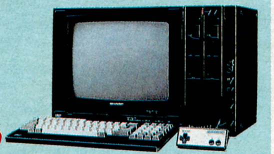
NEW Z-BASIC 驚異の大特価!

X1ターボシリーズ対応 CZ-141SF
(64KB メモリーボード付属)
決算特価 ¥9,800 (〒1,000)



プリンター

CZ-8PC2 カラー漢字熱転写プリンター……決算特価¥49,800
CZ-8PC3 カラー漢字熱転写プリンター……定価¥~~65,800~~
IO-730 カラーイメージ・ジェットプリンター……定価¥~~230,000~~
※その他、他社メーカープリンターも特価販売中。



X1G モデル30セット

- CZ-822CB 本体
- CZ-820DB ディスプレイ
- 人気ゲームソフト
- オリジナルゲームパック
- ディスケット……サービス

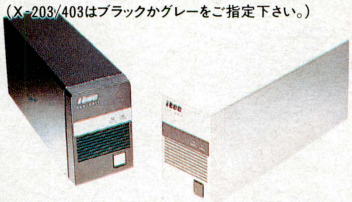
決算特価 ¥79,800
※X1 twinも特別販売中ですヨ!

ツクモでハードディスク

お財布にやさしい価格!

●ウィンテク
HD-404HS(40MB 28ms) 決算特価¥103,800

●アイテック
IT X-203(20MB 28ms)……決算特価¥73,800
IT X-403(40MB 29ms)……決算特価¥99,800
(X-203/403はブラックかグレーをご指定下さい。)



モデム



オムロン MD-2400B
300/1200/2400ボー
決算特価¥39,800

MD-1200A2
300/1200ボー
決算特価¥17,800

通信ソフト

X1 turbo用
SPS JETターボターミナル……決算特価¥8,400

X68000用
SPS た~みのる……決算特価¥10,900

シャープ Communication PRO-68K……標準定価¥~~19,800~~

マウス/トラックボール

(X1、X1turboシリーズ/MZ-2500シリーズ対応)

シャープマウス
CZ-8NM2(A) 定価¥~~6,800~~

シャープトラックボール
CZ-8NT1…定価¥~~13,800~~

ツクモオリジナルマウスセット



TS-MX1+マウスパッド
決算特価 ¥5,500 **限定**

ポケコン/電子手帳



シャープ
PA-8500
定価¥28,000
大型4行表示、テータスケジュール管理に便利。ICカード、プリンタで更に発展するハイグレードタイプ。
決算特価 ¥24,800



シャープ PC-E200
定価¥22,000
決算特価¥17,800



シャープ PC-E500
定価¥28,800
決算特価¥24,800

近日発売! 3.5インチ増設ドライブ

- TS-3X68
- キーボード延長ケーブル

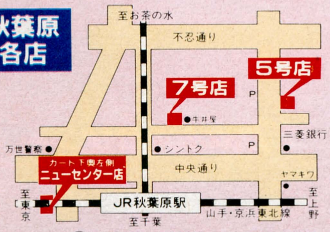
お問い合せは7号店まで……



ツクモは「スーパーX PRO SHOP」です。

PRO STAFF ツクモ

秋葉原各店



㊔AM10時~PM7時 ㊡毎週木曜日

ツクモ7号店	03-253-4199
通信販売部	03-251-9911
ツクモ5号店	03-251-0531
ニューセンター店	03-251-0987
名古屋1号店	052-263-1655
名古屋2号店	052-251-3399
ツクモ札幌	011-241-2299

九十九電機㈱ 〒101-91 東京都千代田区神田郵便局私書箱135号

全国代金引き換え配達
お申し込みは☎03-251-9911へお電話1本!
商品到着の際、玄関でお会計ができます。配達日の指定もできます。

夏・冬、ボーナス2回払い受付中
月々¥3,000以上の均等払いも頭金なし。

現金書留なら
〒101-91 東京都千代田区神田郵便局私書箱135号
九十九電機㈱通信販売部

銀行振込なら
事前に☎でお届け先をご連絡下さい。
富士銀行 神田支店㊜No.894047


Falcom
迫り来る“魔”の波動を感じたか!?
ならば旅立て、
暗黒のギルバレス島へ!!
SORCERIAN SYSTEM
SCENARIO
Vol.3
ピラミッドソーサリアン
Sorcerian System
『戦国ソーサリアン』で暗黒神・邪鬼を倒すことは、“魔”の物語の序章にしかすぎなかった! 暗黒神・邪鬼を凌ぐ強大な“魔”の本体、大魔王ギルバレスに挑む冒険だから、『ピラミッド・ソーサリアン』が、『戦国ソーサリアン』以上のボリュームを持っているのは、当然かもしれないが、それにしてもすさまじい。5インチ2Dディスクでは、やっとのことで2枚に収めたものの、内容の濃さは5枚分!? 1から5までが連なった長編シナリオであるというだけでなく、謎解きあり、アクションありで、各シナリオごとに、その性格が変えられているんだ。これほど彩り豊かで、これほど手ごわいゲームなんて、『ソーサリアン』にもなかったぜ!
SORCERIAN SYSTEM SCENARIO Vol.3
Pyramid Sorcerian
ピラミッド ソーサリアン
好評発売中!
X1turbo 5.2D(2枚組)3,800円
Falcom 日本ファルコム株式会社
Personal Computer Software
〒190 東京都立川市柴崎町2-1-4 トミオービル
通信販売(送料無料)
●現金書留の場合
氏名・機種名・住所・氏名・電話番号を明記して、現金書留でお申し込みください。
●代金引換の場合
電話やFAXやハガキで、品名・機種名・住所・氏名・年齢・電話番号を明記して、お申し込みください。商品お届け時に商品代金をお支払いください。
TEL 0425(27)6501
FAX 0425(28)2714
SORCERIAN
Sorcerian System
SCENARIO Vol. 2
戦国ソーサリアン
好評発売中!!
SORCERIAN SYSTEM
SCENARIO Vol.2
戦国ソーサリアン

超低金利クレジットOK

またまた秋葉原で おなじみの

装いも新たに大きくなった!!
特報!!
中古パソコンの
現金買取・下取OK
(差額は低金利、クレジットをご利用下さい。)

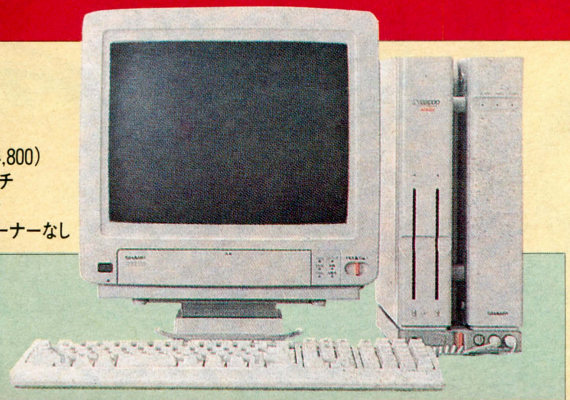
1/15~2/20

●お近くの方はお
●本体単品で特
●ビジネスソフト定

ご利用下さい。
超低金利クレジット

12回	4.5%
24回	9.5%
36回	13%
48回	17%
60回	22%

X68000ACE HD (送料¥2,000)



NEW
CZ-603D
(定価¥84,800)
●0.31ピッチ
●14インチ
●TVチューナーなし

Ⓐセット:CZ-611C+CZ-611D+M-2HD(10枚)+ゲーム
………定価¥544,800➡P&A超特価(価格はお電話下さい)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑセット:CZ-611C+CZ-601D+M-2HD(10枚)+ゲーム
………定価¥519,600➡P&A超特価(価格はお電話下さい)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓒセット:CZ-611C+CZ-603D+M-2HD(10枚)+ゲーム
………定価¥484,600➡P&A超特価

Ⓓセット:CZ-611C+Cu-21CD+M-2HD(10枚)+ゲーム
………定価¥539,600➡P&A超特価(価格はお電話下さい)

※X68000セットでお買い上げの方にゲームの他にドラゴンスピリッツ(¥8,800)をプレゼント!
※チルトスタンド(CZ6ST1 ¥5,800)必要な方は¥5,000加算して下さい。

X68000ACE (送料¥2,000)



X68000用
ジョイスティック
送料¥500
●XE-1PRO
定価¥9500
特価¥7,800
●ASCII STICK X-TURBO
定価¥6,800▶
特価¥5,500

Ⓐセット:CZ-601C+CZ-611D+M-2HD(10枚)
………定価¥464,800➡P&A超特価(価格はお電話下さい)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑセット:CZ-601C+601D+M-2HD(10枚)
………定価¥439,600➡P&A超特価(価格はお電話下さい)

12回	?	24回	?	36回	?	48回	?-	60回	?

Ⓒセット:CZ-601C+CZ-603D+M-2HD(10枚)
………定価¥404,600➡P&A超特価(価格はお電話下さい)

Ⓓセット:CZ-601C+Cu-21CD+M-2HD(10枚)+ゲーム
………定価¥459,600➡P&A超特価(価格はお電話下さい)

※チルトスタンド(CZ-6STI ¥5,800)必要な方は¥5,000加算して下さい。
※X68000セットでお買い上げの方にゲームの他にドラゴンスピリッツ(¥8,800)をプレゼント!

X-1ターボZⅢ/ZⅡ/Z (セットでお買い上げの方にディスケット10枚 ジョイカード、ゲーム3種プレゼント) 送料¥2,000



Ⓐセット:NEW
X-1ターボZⅢ(CZ-888C+CZ-860D)
定価¥269,600▶価格はお電話下さい

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑセット:NEW
X-1ターボZⅢ(CZ-888C+CZ-830D)
定価¥269,600▶価格はお電話下さい

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓒセット:X-1ターボZⅡ
(CZ-881C+CZ-880D)
定価¥289,600▶特価**¥182,000**

Ⓓセット:X-1ターボZ
(CZ-880C+CZ-880D)
定価¥327,800▶特価**¥158,000**

X-1 TWIN/G (送料¥2,000)

Ⓐセット:X-1TWIN(CZ-830C+CZ-820D)…………定価¥179,600▶特価**¥94,000**
Ⓑセット:X-1Gモデル30(CZ-822C+CZ-820D)……定価¥197,800▶特価**¥79,000**
●セットでお買い上げの方に、ディスケット10枚、ジョイカード、ゲーム3種、パソコンラックⒶ3段をプレゼント

プリンターセット ※全セットにケーブル、用紙付 (送料¥1,000)

Ⓐセット:CZ-8PC2 限定……………………定価¥69,800▶特価**¥44,000**
Ⓑセット:CZ-8PC3……………………………定価¥65,800▶P&A超特価(お電話下さい)
Ⓒセット:CZ-8PK7…………定価¥122,000▶P&A超特価(お電話下さい)

12回	?	24回	?	36回	?

Ⓓセット:CZ-8PK8…………定価¥152,000▶P&A超特価(お電話下さい)

12回	?	24回	?	36回	?

Ⓔセット:CZ-8PK9…………定価¥89,800 ▶P&A超特価(お電話下さい)

12回	?	24回	?

Ⓕセット:CZ-8PK6……………………定価¥159,000▶超特価**¥69,000**
限定品、用紙1,000枚付、送料無料

お知らせ 11月1日より、P&A本店の住所が変わります。装いも新たに大きくなりました。今後とも、ご愛顧下さるようお願い致します。

これが僕のステイタス。

あれこれ迷っている人、すぐウェーブ・アイへTELして下さい。

お申し込みは料金無料のフリーダイヤルで

0120-009898

だから安心

ウェーブ・アイ10（テン）ポイントチェック

チェック1 全品2倍保証！
メーカー保証の2倍の保証がついています。（メーカー1年保証＋ウェーブ・アイ1年保証）

チェック2 夏のボーナス一括払いOK！
商品は今すぐお手元に、お支払いはまとめて夏のボーナスで!!

チェック3 超低金利クレジット
3回〜72回までのクレジットが格安の金利でOK。また当社提示支払い例のほかにお客様独自の支払いプランが組めます。

チェック4 商品先取り、支払いは半年先から。
支払い開始は半年先！でも商品はほしいというお客様でもOK！

チェック5 ボーナス2回払いOK！
月々の支払いは全くナシ！お支払いは冬と夏のボーナスでOK！

チェック6 代金引換OK！
現金一括にしたいというお客様、お支払いは現品到着時でOK！（但し離島の方はご利用できません。）

チェック7 全国無料配送
一部地域を除き送料無料で商品をおとどけします。（但し5万円以上の商品に限ります。離島の方は有料となります。）

チェック8 配達日指定OK！
留守がちの方の為に、ご都合に合わせて、配達致します。もちろん日曜・祭日もOK。

チェック9 下取り、買取りもOK!
お手持ちのパソコンを下取りして、わずかな予算で新製品と買い換えることができます。

チェック10 ハガキ注文もOK！
いそがしくて電話をするひまが無いという方の為に、ハガキでのご注文もOK！

〒252
藤沢市湘南台
一丁目一〇番地一号
ウェーブアイ
Oh！X係

1. 住所
2. 氏名
3. 年令
4. 電話番号
5. 保護者名 ㊞（20才未満の方）
6. 商品名
7. 支払い方法
月々 円× 回
ボーナス 円× 回



X68000 ACE HD

X68000に20MBハードディスクを搭載。
ますます熱くなるクリエイティブ＆パーソナルワークステーション。

プラン249 X68000ACE HD お買得基本セット TELにて

CZ-611C	399,800円
CZ-603D（0.31ミリ、高解像度CRT）	84,800円
ブランクディスケット3M（5インチ2HD）10枚	23,000円
定価合計	507,600円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

7,000円×36回	ボーナス27,300円×6回
5,000円×48回	ボーナス23,800円×8回
3,000円×60回	ボーナス26,800円×10回
6,400円×72回	ボーナス なし

プラン250 X68000ACE HD 純正基本セット TELにて

CZ-611C	399,800円
CZ-601D（0.39mm、TV内蔵CRT）	119,800円
ブランクディスケット3M（5インチ2HD）10枚	23,000円
定価合計	542,600円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

8,000円×36回	ボーナス26,900円×6回
5,000円×48回	ボーナス28,200円×8回
4,000円×60回	ボーナス24,500円×10回
6,900円×72回	ボーナス なし

プラン251 X68000ACE HD アートセット TELにて

CZ-611C	399,800円
CZ-603D（0.31ミリ、高解像度CRT）	84,800円
CZ-8PC3（80桁、カラー熱転写プリンター）	65,800円
Z's STAFF PRO-68K（グラフィックツール）	58,000円
PRINT-SHOP-PRO68K（高性能ポップアートシール）	19,800円
CZ-6TU（RGBシステムチューナ）	35,800円
CZ-6VT1（カラーイメージユニット）	69,800円
A4カット紙100枚	480円
ブランクディスケット3M（5インチ2HD）10枚	23,000円
定価合計	757,280円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

12,000円×36回	ボーナス40,800円×6回
10,000円×48回	ボーナス27,600円×8回
8,000円×60回	ボーナス25,000円×10回
10,500円×72回	ボーナス なし

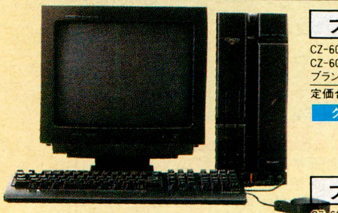
プラン252 X68000ACE HD ミュージックセット TELにて

CZ-611C	399,800円
CZ-601D（0.39mm、TV内蔵CRT）	119,800円
CZ-8PC3（80桁、カラー熱転写プリンター）	65,800円
MUSIC PRO-68K（楽譜ワープロ）	18,800円
SOUND PRO-68K（FM音源サウンドエディタ）	15,800円
ED-700（2段パソコンラック）	27,000円
A4カット紙100枚	480円
ブランクディスケット3M（5インチ2HD）10枚	23,000円
定価合計	670,480円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

12,000円×36回	ボーナス26,800円×6回
9,000円×48回	ボーナス22,700円×8回
7,000円×60回	ボーナス22,000円×10回
9,200円×72回	ボーナス なし



X68000 ACE

ハードの余裕がフレンドリーなオペレーションを生みだしている。ますます熱くなるクリエイティブワークステーション。

プラン246 X68000ACE純正お買得基本セット TELにて

CZ-601C	319,800円
CZ-603D（0.31ミリ、高解像度CRT）	84,800円
ブランクディスケット3M（5インチ2HD）10枚	23,000円
定価合計	427,600円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

10,000円×24回	ボーナス24,800円×4回
6,000円×36回	ボーナス22,300円×6回
4,000円×48回	ボーナス21,300円×8回
6,200円×60回	ボーナス なし

プラン247 X68000ACE純正基本セット TELにて

CZ-601C	319,800円
CZ-601D（0.39mm、TV内蔵CRT）	119,800円
ブランクディスケット3M（5インチ2HD）10枚	23,000円
定価合計	462,600円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

6,000円×36回	ボーナス26,100円×6回
4,000円×48回	ボーナス24,200円×8回
3,000円×60回	ボーナス22,200円×10回
5,700円×72回	ボーナス なし

プラン248 X68000ACEミュージックセット TELにて

CZ-601C	319,800円
CZ-603D（0.31ミリ、高解像度CRT）	84,800円
CZ-8PC3（80桁、カラー熱転写プリンタ）	65,800円
Music PRO68K（簡単操作の楽譜ワープロ）	18,800円
Sound PRO68K（FM音源サウンドエディタ）	15,800円
Sanpling PRO 68K（高機能サンプリングエディタ）	17,800円
AN-160SP（アンプ内蔵スピーカーシステム）	59,800円
A4カット紙100枚	480円
ブランクディスケット3M（5インチ2HD）10枚	23,000円
定価合計	606,080円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

10,000円×36回	ボーナス29,400円×6回
7,000円×48回	ボーナス27,400円×8回
5,000円×60回	ボーナス27,900円×10回
8,300円×72回	ボーナス なし

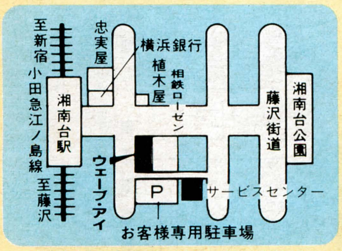
夏の 商品今すぐ!! ボーナス一括払いOK

受付時間 AM9:30▶PM9:00 電話一本即納！

藤沢	0466(43)1775	静岡	0542(54)0696
札幌	011(771)4971	名古屋	052(581)4325
盛岡	0196(24)3172	大阪	06 (362) 5057
仙台	022(267)5371	広島	082(293)0811
新潟	0252(75)5076	福岡	092(481)0502
東京	03 (226) 9286	FAX	0466(43)1265

18歳未満の方は保護者とご一緒にお電話下さい。

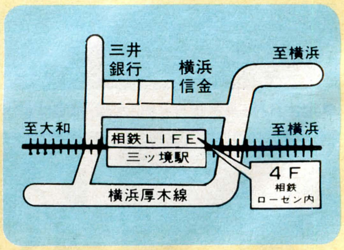
新歩人の情報ターミナル



湘南台店☎0466-43-1771

未来をクリエイトする
WAVE EYE
ウェーブ・アイ

〒252神奈川県藤沢市湘南台1−10−1
振込銀行▶ 横浜銀行 湘南台支店 当座000467（株）ウェーブ・アイ
第二・第三火曜日定休日



三ッ境店☎045-363-7044

コンピュータのイメージとマシン語

マシン語とはいったいどんなものなのでしょうか。どうしてマシン語を使えばハードウェアの能力を100％引き出せるのでしょうか。これらの答えは，マシン語についてある程度わかってからでないと，正確に理解することは難しいと思います。そこで，ここでは直観的な説明をしておきたいと思います。

図1を見てください。これはユーザーから見たコンピュータのイメージです。図の中央にあるのがハードウェアで、それをソフトウェアのいくつかの層が重なりあって包んでいます。そして，この外側に私たちユーザーがいます。つまり，ユーザーの位置から直接ハードウェアの部分は見えないことになります。これはユーザーがハードウェアを直接操作できないことを示しています。

そこでユーザーは，BASICや他の言語でプログラムを書くわけです。また図中の矢印は，各層が矢印の方向に指示の内容を伝達するというアプローチを示しています。つまりこの図の一番外側の矢印が示すように，ユーザーはアプリケーションプログラムを書くというアプローチを行い，やりたいことの指示を行うわけです。するとBASICの処理系は，私たちに代わってハードウェアを操作します。

ここで注意してほしいのは，私たちユーザーの指示が直接ハードウェアに行くのではなく，いくつかの層を通り，指示の形が変わりながら（指示の内容は変化しない），ハードウェアに伝達されているということです。

では，直接ハードウェアに対して指示を伝達することはできるのでしょうか。実は，一番内部のハードウェアはマシン語で書かれたプログラムしか解釈（実行）できません。つまり，マシン語はハードウェアを直接操作できる唯一の言語なのです。

一方，ハードウェアから見たコンピュータのイメージは図2のようになるでしょうか。マシン語を使えるようになるためには，コンピュータがどのように構成されているかということ，すなわち“コンピュータのアーキテクチャ”についての知識はもちろん，各ハードウェアについての知識，そしてマシン語の知識とそのプログラミングテクニックといったかなり多くの知識が必要です。これはBASICのプログラミングを修得するために必要な知識に比べれば，かなり膨大です。確かに，初心者には難しすぎるかもしれません。それゆえにマシン語が使えるということは，そのユーザーがかなりの実力を持っているという“あかし”であり，他のユーザーの憧れの的となるわけです（本当かなあ）。

(H.S.)

図 1

ユーザーから見た
コンピュータのイメージ

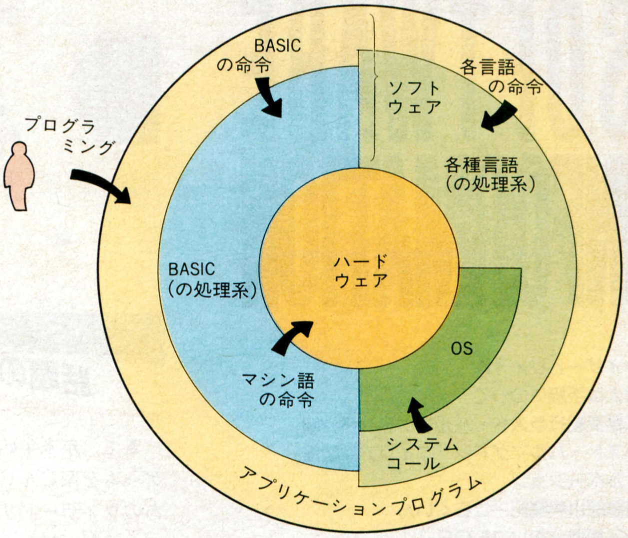
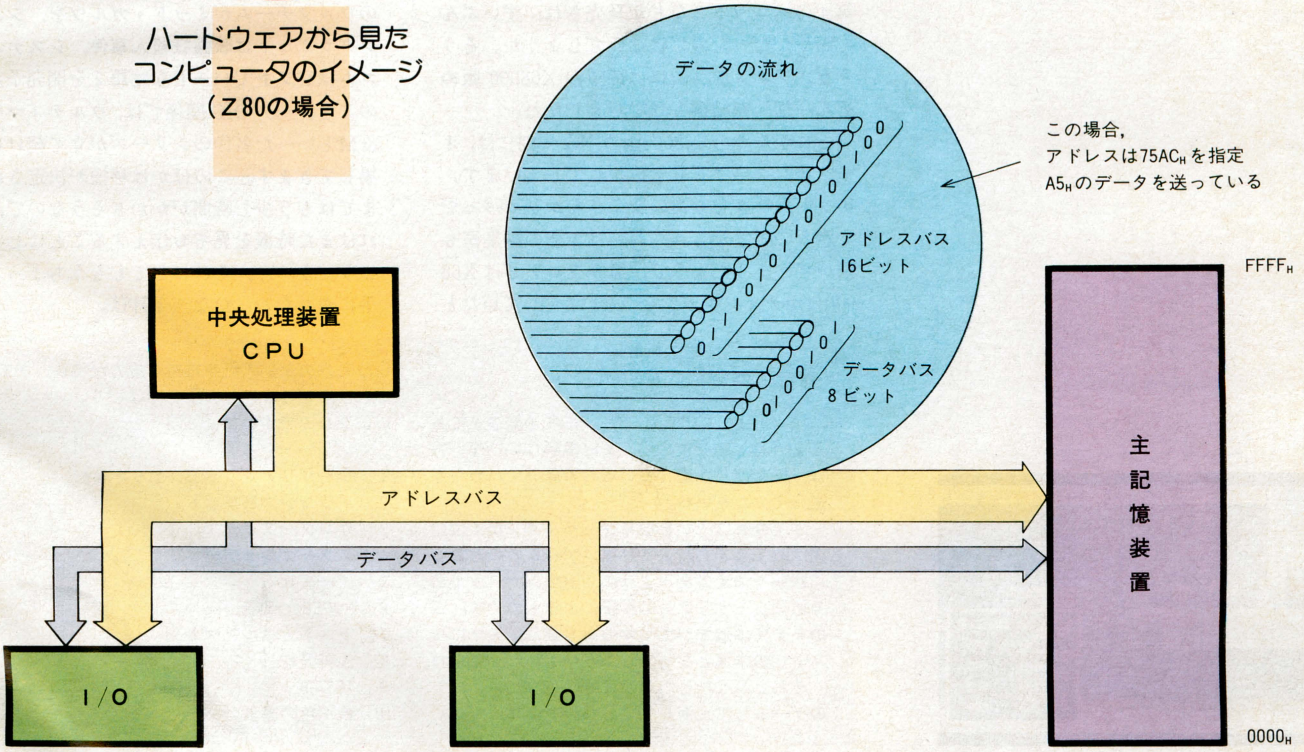


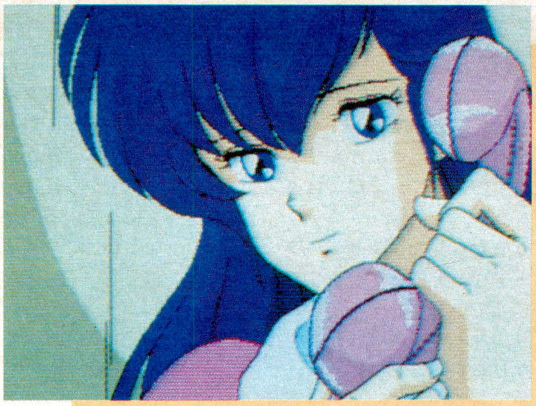
図 2

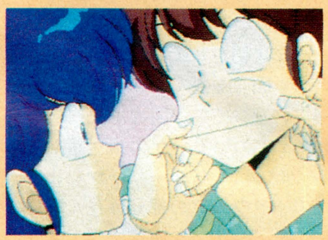
ハードウェアから見た
コンピュータのイメージ
（Z80の場合）



SOFTWARE INFORMATION

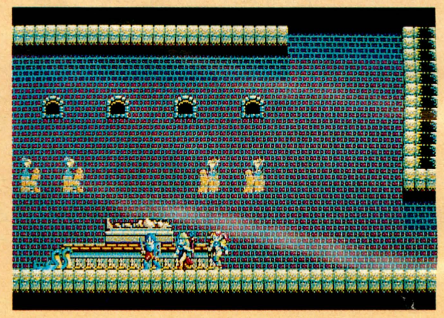
ウィザードリィ＃4
殺人は手紙にのって
妖怪変紀行ラスト・ハルマゲドン番外編
ラスト・ハルマゲドン「エイリアン図鑑」
サルベーション
艶談徳川興隆記・ごらくいん
商品相場分析ソフトGS-1





こちらが新着X68000版のめぞん一刻・完結編に登場の響子さんと五代君。下は発売されたばかりのX1版新九玉伝とピラミッドソーサリアンです





話題のソフトウェア

さて，年末年始に皆さんはいったいどんなゲームを楽しんでいたのでしょうか。X1にはあのウィザードリィ＃4 を始めとしてサイオブレード，新九玉伝といった面々が，またX68000にはMurder Club DXや，パワーリーグ，第４のユニット２などが発売され，ずいぶんと賑やかだったのではないでしょうか。そうそう，アニメファンには待望の X68000 版**めぞん一刻・完結編**も登場しましたね。

とにかくこのX68000版めぞん一刻には，オープニング早々ド肝を抜かれてしまいます。ナント，いきなりあの響子さんがしゃべってくれるのですから。これにはよその編集室も驚いてしまったようで，違うフロアではX68000にボディソニックをつないで聞いていたという，奇特な響子ファンまで現れる始末。ほんとにX68000のおかげで，ゲームの楽しみ方までが変わってしまったようです。

話は変わって，本来ならここで今年前半に発売される新作情報をドッとお届けする予定でしたが，またもや全体的な発売時期のズレ込みが生じてしまったようで，今回は不発。

そんななかで現在までにわかっているのが，X68000ではアークスの移植が決まったばかりのウルフチームが**ミッド・ガルツ**を，シンキングラビットは**映画狂殺人事件**，システムサコムは**ソフトでハードな物語 2** を開発予定とのこと。一方のX1関係では，ウルティマやM＆Mといった名作のシリーズが立て続けに登場してきます。このほかは移植が決定されるまではもう少し時間がかかりそうなので，それはまた時機を見てお伝えすることにしましょう。うーん，それにしてももうちょっと派手にやりたかったな，今月は。

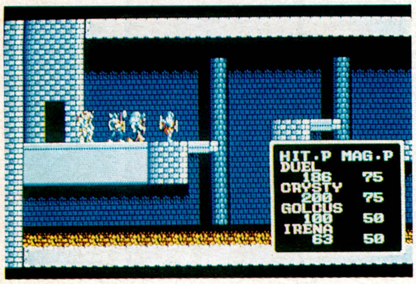
読者が選ぶゲームベスト10

いよいよ，GAME OF THE YEARの発表まであと２カ月となりました。まだ投票していない方は，締め切りが迫っているので急いでハガキを送ってくださいね。

さて今月は，いきなりソーサリアンが１位に復活してきました。これは，ゲーム名のところに説明を加えてあるように，追加シナリオである戦国ソーサリアンの影響によるものです。このままでは追加シナリオのある限り，完全にベスト10の常連となってしまうかもしれませんね。

それともうひとつの注目株はTETRIS。このテのゲームって上海を見てもわかるように，ジリジリと息の長い人気を維持するのが特長。さて，これからどこまでその票を伸ばしていくか，大いに楽しみですね。

1．ソーサリアン（追加シナリオ含む）
2．ドラゴンスピリット
3．サンダーフォースII
4．スタークルーザー
5．イースII
6．SUPER大戦略
7．ラスト・ハルマゲドン
8．沙羅曼蛇
9．TETRIS
10．琥珀色の遺言



新作ソフト情報

☆……12月25日現在発売中　★……近日発売予定

☆ウィザードリィ＃4

あのRPGの代表作，ウィザードリィ第4弾「The Return of Werdna」がついに登場した。このゲームは，ウィザードリィのシナリオⅠで6人のトレボー大王の剣士たちによって倒された悪の帝王ワードナの復活から始まる。そしてなんと今回のシナリオでは，プレイヤーはワードナとなり禁断のアミュレットを手にしたトレボーの大王に永遠の呪いをかけるために旅立つというストーリーのもとにゲームは進められる。ワードナはその魔力によって，味方のモンスターを召喚しながら地上へと向かう。そこにはアミュレットによって偉大な力を持ったトレボー大王が待ち受けているのだ。そうして再びここに壮絶な戦いが繰り広げられる。とにかく，いったいなにが起きるのかまったく謎につつまれたこのゲーム，当然，今回のゲーム構成は上級者向けとなっており，真のエンディングにたどり着けば“Grand Master Adventure”の，そしてそのほかのエンディングの場合には“Master Adventure”の称号が与えられることになっている。なお，このゲームをプレイするためには，これまでのシナリオと違い，シナリオⅠのゲームディスクやキャラクターを持っていなくてもプレイが可能となっている。

X1/X1turbo用　5″2D版3枚組　12,800円
アスキー　☎03(486)8080

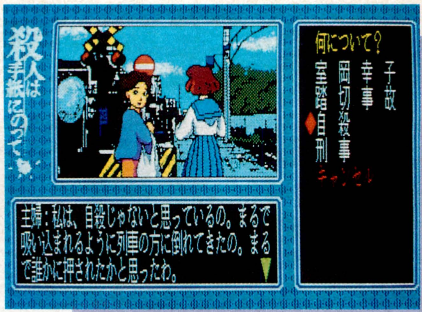
☆殺人は手紙にのって

山崎弘美は17歳。高校生でもあり，そして彼女の持つアパート「コーポ山崎」の管理人でもある。その彼女の元に，大岩刑事が不幸な知らせとともにやって来た。彼女の親友の室岡幸子が自殺したというのだ。しかし，彼女にはそれが信じられなかった。幸子が自殺するにはあまりにも動機があいまいすぎるのだ。そこで，大岩刑事とともに調査に乗り出した弘美なのだが，なんとそこには第2の殺人事件が待ち受けていた。こうして繰り広げられる難事件に立ち向かうAVGなのだが，シーン数50，画面数80以上のストーリーがコマンド選択式でテンポよく進められる。また，初心者のために袋とじのヒント集のオマケ付き。

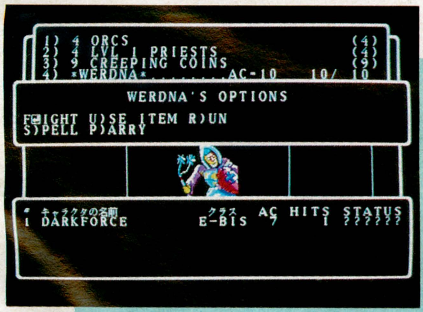
X1turbo用　5″2D版3枚組　7,800円
ツインソフト　☎03(360)3912

★妖怪変紀行ラスト・ハルマゲドン番外編

「最笑最惡の超不可思議RPGついに登場」というキャッチフレーズで登場するのが，この「妖怪変紀行ラスト・ハルマゲドン番外編」だ。当然，このゲームは前作のラスト・ハルマゲドンの番外編という設定なのだが，ストーリーはなぜか地球滅亡の危機を知り日本の妖怪たちが宇宙へと飛び立つところから始まる。しかし，彼らの行き着いた先の星はもうすでに西洋妖怪に支配されていた。その先着の西洋妖怪に次々と無理難題を押し付けられる日本妖怪たち。果たして彼ら日本妖怪たちに明るい未来はあるのか!?　というハチャメチャなノリのゲームなのだ。なお，このゲームのシステムは，ほぼあのラスト・ハルマゲドンと同じだが，あらゆる点でユーザーの意見を取り入れた新しいスタイルのゲームになるらしい。



殺人は手紙にのって



ウィザードリィ＃4

X1turbo用　5″2D版ディスク枚数未定　6,800円
ブレイングレイ　☎03(230)0664

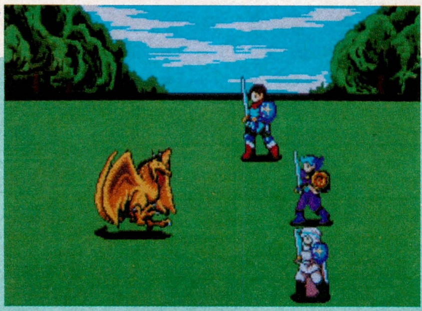
☆ラスト・ハルマゲドン「エイリアン図鑑」

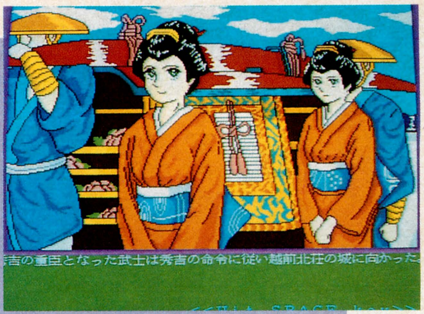
ブレイングレイから発売されているRPG，「ラスト・ハルマゲドン」に登場したエイリアン全150種類の，その強さ，体力，特性などを記したエイリアン図鑑が発売されることになった。さらにこのディスクにはラスト・ハルマゲドン本編のセーブディスクをコピーできるユーティリティと，本編の3D全マップも収録されている。なお，このソフトは通販のみでの販売となっているので，入手方法については迷惑のかからないように直接問い合わせてほしい。またユーザー会員と一般ユーザーとでは価格に違いがあるのでご注意を。

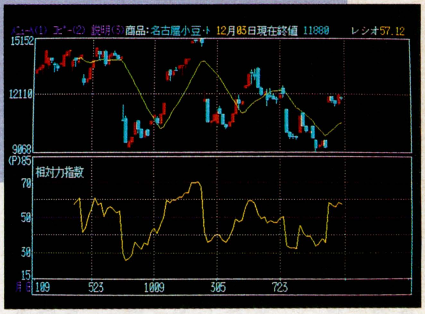
X1/X1turbo用　5″2D版　通信販売のみ
一般ユーザー2,000円/ユーザー会員1,000円
ブレイングレイ　☎03(230)0664

☆サルベーション

ここはどこなのだろう？　私は小鳥のさえずりにほのかな快さを感じながら街のなかを散歩していたはずだったのだ。しかし，あの稲妻は突然，地を裂き，もの凄い勢いで私をどこか遠くへ運んでしまったようだ。目の前の風景はさっきまであった街のそれではなく，見渡す限りの大草原だった。そして，目の前には白く輝く1本の剣が刺さっていた。すべては謎に包まれたまま，その舞台は幕を開けようとしていた……。このようなストーリーで始まるファンタジーRPG「サルベーション」。そのほとんどの操作がテンキーとスペースキーでできる親切設計。魔法やアイテムも豊富。


サルベーション


艶談徳川興隆記・ごらくいん（写真はX68000版）



商品相場分析ソフトGS-1

X1/X1turbo用　5″2D版2枚組　7,800円
全流通　☎06(761)5271

☆艶談徳川興隆記・ごらくいん

全流通から発売のAVG「艶説源平争乱記・いろはにほへと」の続編として登場するのが，この「艶談徳川興隆記・ごらくいん」。前作で武士と静香は平安末期から鎌倉時代初期にタイムトラベルをして，源頼朝と義経の歴史的な地位を入れ替えてしまった。しかし，ここで問題が起きた。鎌倉時代で歴史を狂わせてしまったために，その後の日本の歴史が徐々に狂ってしまい，ついに歴史上では300年近く続いたはずの江戸時代が，なんと10年しか続かないという結果になってしまった。武士と静香はそんな事態を回避するため，今度は徳川幕府の江戸時代へと向かうのだった……。追ってX68000版も発売される。

X1/X1turbo用　5″2D版2枚組　6,800円
全流通　☎06(761)5271

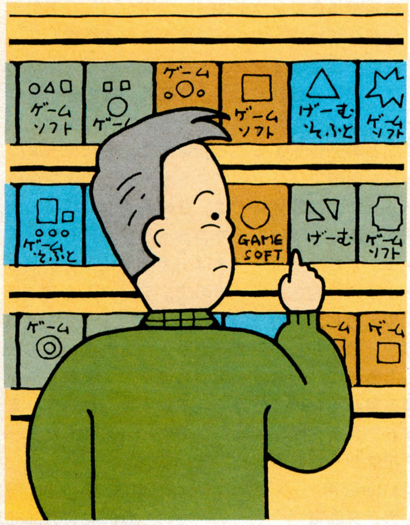
☆商品相場分析ソフトGS-1

株価分析ソフトでお馴染みのテラダ商電から，今度は商品相場分析ソフトが発売された。このGS-1は，株価分析と同じように商品相場の分析をローソク足，相対力指数チャート，相対力指数レシオなどの分析をすることが可能で，分析データの一覧と同時にプリントアウトも可能。このなかにある相対力指数分析では，分析画面中に簡単な説明が加えられていて，マニュアルレスでも使用できる。また，使用データは1週間単位で分析できるように設定されているので，データ入力は週単位で入力し，合計120週分のチャートが表示できるほか，最初から一部データが入力されているものもあるので，それを利用すればそのままデータを積み重ねるだけで使用できるようにもなっている。なお，テラダ商電では，このGS-1を利用したデータ提供サービスも実施している。

X1turbo用　5″2D版　9,000円
テラダ商電　☎0542(78)8662

GAME REVIEW

今月は，X68000には野球ゲームの「パワーリーグ」を，そしてX1には麻雀ゲームの「POWERFULまあじゃん2」と，アドベンチャーの「サイオブレード」をというわけで，なんの脈絡も因果関係もない超豪華メンバーでお届けします。



パワーリーグ

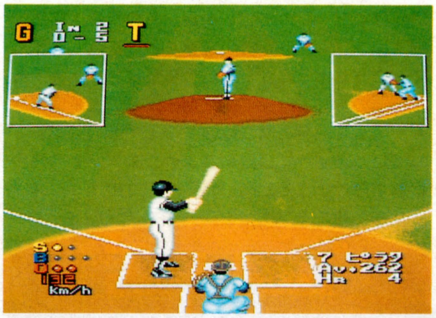
「アウト」，「セーフ」と判定の声も賑やかに，X68000に待望の野球ゲームの登場です。いざ，プレイボール。

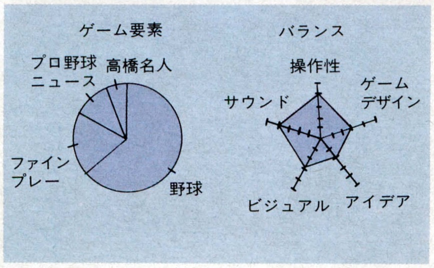
▶たぶん，X68000では初めての実戦野球ゲームだと思うけど，まるで，ファミコンのような野球が楽しめる。といっても，別にけなしているわけではない。パソコンゲームっぽくないということである。ファミコンどころか，そのへんのパソコン野球ゲームなんかはおよびもしない，画面＋「ストライク！」という高橋名人の声などは，誰が見てもX68000なのである。が，しかし，操作はやはりファミコンやPCエンジンのソレなのである。従って，プレイはジョイスティックよりも，ジョイカードのほうが望ましい。

パソコン版の野球ゲームというのは，できるだけシミュレーション性を高めようとするのが普通である。しかし，このゲームは「いかにしてゲーム性を高めるか」をポイントに，しかもあっさりとまとめている。ファミコン（PCエンジン）の持つエンターテイメント性と，X68000のアミューズメント性がよくマッチしたゲームといえよう。

熱中度▶▶▶▶▷▷▷ (M.Y.)

▶いわずと知れたPCエンジン用「パワーリーグ」の移植版である。もちろんX68000用ということでグラフィックやサウンドが強化され，しっかりとAD PCMで音声出力もやっている（声の主が高橋名人なのはご愛嬌)。この種のゲームでは野球が持つ本来の豊富なアクションをいかに簡単な操作で







行えるかがカギになるが，もともとパッドを使うことを考えて作られているだけに，さすがに操作性はいい。ただ打球を追うシーンでは球場の真上からの視点になってしまうので，外野手の配置をすぐには把握できないのがツラいところ。しかしそれでも野球ゲームは面白い。X68000でこのタイプの野球ゲームは初めてなので，X68000ユーザーは揃えておいていいソフトだろう。

ところで，試合終了後のニュースの場面に出てくる女性キャスターを，どうせなら中井美穂サンにもっと似せてほしかったと考えるのは私だけですか？

熱中度▶▶▶▶▶▷▷ (R.Y.)

X68000用 5″2HD版 7,800円
ハドソン ☎011(841)4622

今夜も朝までPOWERFULまあじゃん2

顔面登録や世界トーナメントなど，あらゆる趣向を盛り込んで，前作を遥かに凌ぐ明るさで迫る麻雀ゲームなのです。

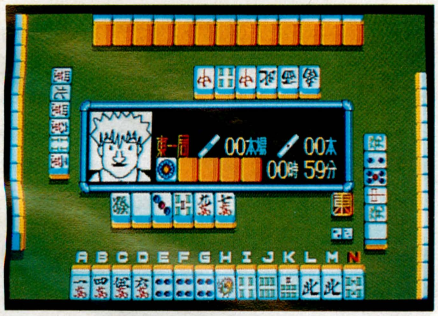
▶おおおおお，モンタージュにスゴロク世界一周に，ポンジャンときたか。ヘンなやつー。これはどういうことかというと，このゲームにはノーマル，エキサイト，さすらい，ぽこ麻雀という4つのモードがあって，このあとのほうの2つがまさしくヘンだったりするわけだ。まずはゲームを始めるときにプレイヤーの顔をモンタージュで作るわけだが，まー，ヘンな顔ばっかできるできる。それから，さすらい麻雀モードなのだが，まあ，これがわけわからん。ロシアンルーレットに競馬にボケクイズときたもんだ（え，わかんない？ そもそもこの狭いスペースでこのゲームの説明をするのに無理があるんだってば)。んで，最後のぽこ麻雀なんだけど，これがまー，平たくいっちゃえばポンジャンなんだけど，アタったときの画面はド派手だし，負けた相手のコメントは笑えるしで，とにかく面白い。座布団1枚あげたいくらい。でも，面

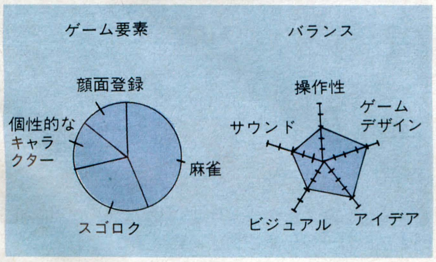
白いんだけど，やっぱりヘンなんだよなー(私はこういうヘンなの大好きだけど)。

熱中度▶▶▶▶▷▷▷ (で)

▶ひと言でいってしまえば，これは麻雀ゲームです。でも，ここに「ただの」と前フリが付かないのがこのゲームの凄いとこ。前作は4つのジャンルからそれぞれ好きなタイプの麻雀が選べるというのが売り物で，その中身といえば「ぽこ麻雀」が比較的目新しいだけ。それ以外はごく普通の麻雀ゲームの集合体といってもだーれも怒らないようなソフトだったのです。でも，どうしちゃったんでしょうね，今回のこのノリは。麻雀ゲームの素材にオマケをぼこぼこ付けて，これでもか，これでもかの大サービス。顔面登録させてプレイヤーの顔を勝手に決めさせるわ，オヂサン相手の4人打ちはあるわ，お姉さんはやたら個性的でベッピンになるわ，ぽこはトーナメント戦だわで，やたらと対戦相手の個性にこだわって，アノ手，コノ手で楽しませてくれます。これを面白いと判断するか，まったく進歩がないと判断するかは自由だけど，わたしゃ前作よりもずっと好きですよ，このゲーム。







熱中度▶▶▶▶▶▷▷ (T.S.)

X1turbo用 5"2D版4枚組 7,800円 (2ドライブ専用)

デービーソフト ☎011(251)7462

サイオブレード

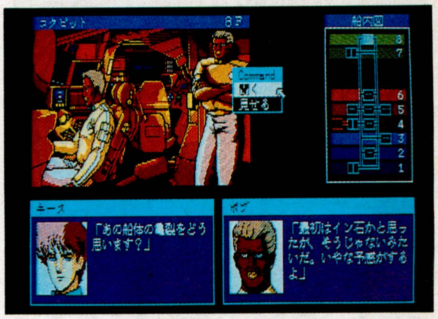
随所にアニメーションが見られて，戦闘モードもイカしてる。なかなかに見せてくれるアドベンチャーの最新作です。

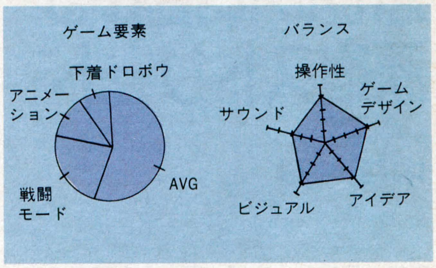
▶雰囲気はかなりいいみたいだ。オープニングを見たおいらはゾクゾクしちゃいましたよ。主人公が2人いてひとりは宇宙で，もうひとりは地上で活躍というのもなにか映画を思わせるようでよろしいですね。音楽もかなり状況にマッチしていて，ゲームを緊迫感あふれるものにしていますよ。メロディモジュールなるものがパッケージに付属していて，これがゲームを解く鍵になっています。こういうゲームにヒントを与えるような「おまけ」というのはなかなか新しい。まだ，おいらは解き終わっていないけど，ディスク5枚組で1枚1枚クリアしていくのがとても楽しみ。ビデオにアニメを撮り溜めしておいたのを週末にまとめて見る，みたいな感じで次のディスクへの期待感がたまらないよん。まあ，欲をいえば漢字を16×8の圧縮漢字にしないで(おいらは圧縮漢字が大大大大嫌い!) ちゃんとした漢字を使ってほしかったなぁと思うけど，最近遊べるソフトの少ないX1ユーザーは買い! のソフトかも。うーまーいーぞー!!

熱中度▶▶▶▶▶▶▷ (善)

▶アニメーションが売りのアドベンチャーです。エレベーターのドアが開くという子供だましのものから，女性のシャワーシーンなんて大人だましのものまで，さまざまなアニメがふんだんに盛り込まれています。アニメで動きを自然に見せるためには，毎秒8コマの動画が必要だそうですが，さすがにそれには及ばず，少々ぎこちない感じはします。とはいえ，絵がしっかりしているので見応えはあるでしょう。







ゲームは，まずマウスやカーソルキーで画面上の物を指定することから始めます。この方式は視覚的直観的でわかりやすいとともに，どこかに取りこぼしがあるのではないかという不安が常につきまといます。しかしいまのところ，これがベストかな?

以上のように，アニメも効果的だし，操作性もいいのですが，前半のストーリーが一本道ともいえる展開で少々弱いのが残念です。後半は……，あとひと息のところで行き詰まってしまいました。いやはや申しわけない。

熱中度▶▶▶▶▶▷▷ (お)

X1turbo用 5"2D版5枚組 8,800円 (2ドライブ専用)

ティーアンドイーソフト ☎052(773)7770

スポーツゲームが欲しいっ!

ないものねだりをするのは本当はいやなんだけど，やっぱりいっちゃう。なんで，最近のゲームにはスポーツゲームが少ないんだよー。最近出たそのテのゲームって，「パワーリーグ」に「熱血高校ドッジボール部」だけじゃないかー，もう，プンプン。

だいたい，スポーツゲームほど単純で楽しめて，作ろうと思えばいくらでもバリエーションを増やすことができるジャンルはそうないのに，どうして作らないの? スポーツっていえば野球ゲームばかりじゃないの，最近のX1やX68000に出てるゲームって。

やっぱり私としては，ここでひとつテニスゲームの新しいのが欲しいな。ファミコンの任天堂のテニスぐらいの単純なやつでいい。シンプルなゲームのほうがのめり込めるから。そうそう，それにオプションで紫外線電球を付けて，ゲーム中ずっとCRTの横につけておけば，たちまちテニス焼け少年の一丁上がり! うーん，かっこいい(よくない，よくない!)。 (で)

THE SOFTOUCH

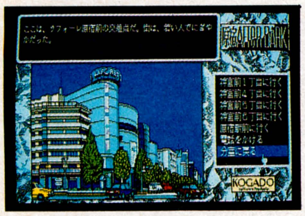
●原宿アフターダーク



悲しみに揺れる原宿ストリート

Nakamori Akira
中森　章

東京・原宿といえば，華やかな若者の街。その街の裏側で展開される悲しい人間模様。それらをうまく取り入れたAVGが，このタケルソフトの原宿アフターダーク。これぞ本格派推理ミステリーと呼ぶのにふさわしいゲームなのかもしれません。



X1turbo用　5"2D版3枚組　6,500円（2ドライブ専用）
ブラザー工業　☎052(824)2493

FILE 0:初めての原宿

原宿か，そういえば長い間東京に住んでいるけど，原宿には一度も行ったことがありません。原宿といえばファッションの街。何かきらびやかな雰囲気に満ちて，僕みたいな着た切りすずめの青年（おぢさん？）にとっては，どこか異国の地という気がして敬遠してしまうのです。

この「原宿アフターダーク」は，そんなきらびやかな華やかさの裏側にうごめく，人間の愛憎と欲望をテーマにしたアドベンチャーゲームなのです。これは原宿で発生した2件の殺人事件を追って奮闘を続ける刑事の物語です。あなたは原宿分署の刑事となって聞き込みを続けて行くわけですが，練りに練られたシナリオの前に思わず唸ってしまうに違いありません。

それとこのゲームは，コマンド選択式を採用していますが，それをキーボード，マウスあるいはジョイスティックの3つから選べるのが新鮮です。また，このたぐいのゲームは，何時間もかけて機械的にすべての組み合わせを試していけば解けるものもありますが，この「原宿アフターダーク」はその心配はありません。先に進めば進むほど登場人物が増えて，それがなんらかの事件と直接的あるいは間接的に関わってくるのです。

ですから，進めば進むほど世界が広がってきて，本当にこれらの謎が収束するのだろうかと不安さえ感じてしまいます（ホントいうと，このゲームを数日で解いてと編集室でいわれたときにはアセッてしまった)。しかし行き詰まったという感じはほとんどなく，原宿という限られたスペースのなかに壮大な物語を詰め込もうとする，開発者の意地みたいなものをヒシヒシと感じることができるのです。

このゲームと感じのよく似たアドベンチャーゲームとして，リバーヒルソフトのJ.B.ハロルドシリーズがありますが，舞台設定がアメリカではなく，こちらは東京は原宿だということもあって，ファッション用語は飛び交うわ可愛い女の子（川久保恵を除く）は大勢出てくるわで，さらっと軽くてより身近なノリで楽しませてくれます。

さらに，このゲームで特筆すべきところに聞き込みモードがあります。コマンドで「聞き込みをする」を選択すると，画面が原宿の地図に変わり，その地図のなかを豆粒みたいな刑事をテンキーで移動させながら，一軒一軒情報を聞いていくことができるのです。そしてこのモードは決して飾りで付いているのではなく，事件関係者の自宅付近ではこの聞き込みによって重要な証言を得ることができるのです。

これは犯罪の捜査は足でということを実感させてくれる素晴しいモードといえるでしょう。なお，この地図は原宿駅前，神宮前1丁目，神宮前4丁目，神宮前5丁目，神宮前6丁目，および出張先の3カ所が用意されていて，行ったことのない原宿の街を疑似体験させてくれるのが嬉しいです（地図がどのくらい正確なのかはわからないのが悲しい）。

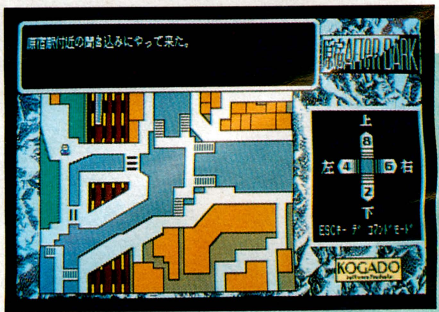
さて，前置きが長くなりましたが，ここらで「原宿アフターダーク」の物語を少し紹介することにしましょう。ゲームの性格上あまり詳しいことは紹介できませんが，物語そのものがかなり複雑ですから，それを整理する目的で事件の何人かの登場人物に関して捜査メモ風にまとめてみることにしましょう。

FILE1:プロローグ

3月25日の朝方，代々木公園原宿側入り口の南側の林のなかで若い女性が死んでいるという110番通報があった。発見者は角谷文太という医者である。朝のジョギングでこの死体を発見した。警察の調べによると被害者は女装をした男であった。腹部を刃物でえぐられているわりには，現場の草むらにほとんど血が付いていないこと，死体を引きずった跡があることなどから，別の場所で殺されて運ばれてきたらしい。現場からは，血の付いた婦人物のジャケット，そして木にひっかかっていた白い布きれ，そしてカツラが証拠品として持ち帰られた。

FILE2:毛利　裕

女装した死体で発見された男は，神宮前6丁目の刈谷不動産の社員である毛利裕であることが判明した。毛利は刈谷不動産の社員として，神宮前1丁目のメゾン原宿の買収を精力的に行っていたらしい。これは香月クリニックの新たな建設用地を確保す



聞き込みは一軒一軒根気よく続けよう

るためだ。ここで彼はかなり強引なやり方をしたらしく，メゾン原宿の住人にはノイローゼになって自殺した人間もいて，毛利に恨みを持つ人は多かったようだ。

毛利の部屋からは6人の名前が書かれた紙切れと3枚の写真が発見された。紙切れにあった名前のうち，5人は問題のメゾン原宿の住人で残りのひとりは毛利に部屋探しを頼んでいた安積麻衣という人だった。写真は看護婦姿の女性，OLスーツ姿の女性およびなにかの細胞のようなものの写真だった。のちほど，OLスーツの女性は毛利の恋人の遠野ゆかり，看護婦姿の女性はゆかりの姉の遠野かおり，細胞は人間の卵細胞であることが判明する。

FILE3:遠野ゆかり

毛利裕が殺された直後，原宿駅前競技場入り口で第2の殺人が発生した。被害者は遠野ゆかり。左胸を刃物でひと突きされたらしく，胸のあたりを中心に大きな血だまりができていた。近くの植木に血が付いた男物のワイシャツの袖が引っ掛かっており，それが証拠品として持ち帰られた。

遠野ゆかりというのは毛利の恋人だった。神宮前5丁目のフルーレというファッション関係の雑誌社のライターである。ゆかりの姉，遠野かおりは8年前に自殺したのだが，ゆかりは毛利と同じように姉の死に方に疑問を抱いて調査していたという。親しい人には「姉の仇を討つために姉が最初に勤めた病院から調べ直す」ともらしていたそうだ。

そこで運よく同様に妹の死因を調査していた毛利と知り合いになり，2人で調査を進めているうちに，偶然にも毛利めぐみと遠野かおりの死には共通の人物が絡んでいることを知ったらしい。

そこでゆかりは，毛利と共謀しその人物の子供を誘拐し，身代金を奪ったあとで子供を殺してしまおうと計画していたが，逆にその人物に殺されてしまったのだ。フルーレ編集部にゆかりの住所を知りたいという女性からの電話があったそうだが，犯人は女性なのだろうか。それと男物のワイシャツの袖との関係はあるのだろうか。


システム手帳がないと電話はかけられない

FILE4:毛利めぐみ

毛利裕の父親の親友に安積という人物がいた。安積は若くして死んだため毛利の父親が残った母と子の面倒を見ていた。ある日，毛利の父親が安積の娘のために買った服を見つけて着ようとし，めぐみと安積の娘とが奪い合いとなり，過って土間に落ちためぐみが足を悪くしてしまった。それ以来，毛利裕は安積親子を恨んでいたという。

毛利は刈谷不動産の社員になる前，モダームというファッション関係の会社に勤めていたが，妹のめぐみもドレープというファッション関係の会社に勤めていた。そこでめぐみは他社のデザインを盗んだというぬれぎぬを浜津に着せられて，ドレープを退社した。その後ファッションメーカーのピンクアリスに就職したが，雅装という会社の従業員，御手洗に乱暴されたのがショックで，彼女は田舎に帰っていた。

FILE5:遠野かおり

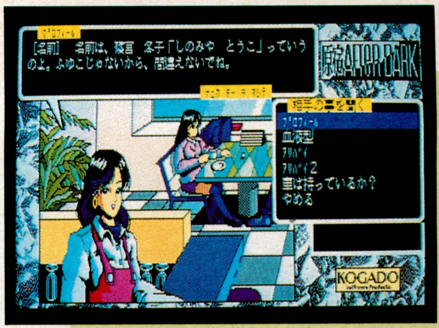
遠野かおりは遠野ゆかりの姉である。神宮前6丁目の君原病院で看護婦をしていたが，数年前刈谷不動産の社長の娘を医療ミスで殺しそうになった。そのせいで君原病院の評判は一時ガタ落ちし，給料が払えなくなったため君原病院をやめさせられた。

その後は君原病院の院長の先輩である角谷文太が院長をしている神南2丁目の角谷病院で働いていたが，ここでもミスを犯して病院を飛び出してしまった。角谷病院の看護婦の話によると，看護婦の仕事が怖くなったので別の仕事をやりたいということだった。新たな勤め先は駅前の喫茶店ベルだったが，アフロスに行くという言葉を残したまま消息を断った。

FILE6:刈谷 祥

刈谷祥は，毛利裕の勤めていた刈谷不動産の社長である。現在は香月クリニックの院長の香月三千夫に頼まれて，メゾン原宿の買収をやっている。刈谷は今の不動産業を始める前はピンクアリスというファッション関係の会社で経理をやっていた。毛利めぐみが一時期勤めていた会社である。ピンクアリスは川久保恵という有能なデザイナーのおかげで好評を博していたが，あるとき粗悪品のコピー商品が大量に出回り評判を落としてしまった。

これはモダームの業田が川久保を引き抜くためにに浜津を使ってやらせたことだった。このときコピーを作ったのが安積麻衣ということらしい。川久保がモダームに移籍しピンクアリスが傾きかけていた頃，刈谷は会社の金を横領して逃げてしまった。


彼女もまた秘密を隠しているようなのだが……


毛利と刈谷は以前ここに勤めていたらしい

このためピンクアリスは倒産し，社長の君原信一郎は責任を感じて自殺している。

FILE7:いけますよ，このAVG

お疲れ様。こんな感じで人物の説明を簡単に並べただけでも，このゲームのストーリーがかなり複雑であることがわかるでしょ。さらに物語はクローン人間の話や愛人バンクの話を絡めてますます複雑な方向へと展開し，最後に見事な収束を成し遂げます。まさかあの人が犯人だったとは……。

そして，その犯人の自供には泣かされます。とにかく，このゲームはアドベンチャーの命ともいえるシナリオの完成度はかなりのものです。これだけの人物と数多くの設定が絡み合いながら，見事にひとつにつながるという離れ技を見せてくれた工画堂スタジオは，これでまたひとつジャンルの幅を広げたような気がします。

いずれにしても，この「原宿アフターダーク」は，X1turboでは久々に“解いた”ということを実感できるアドベンチャーゲームだといえるでしょう。絶対に“買い”ですね。それにしても，業田社長の妻の貴子が「ホトトギス」という花を示して犯人を教えようとしていたとは，最後の最後まで説明を聞かないとわかりませんでした（普通はわからないよねえ）。

THE SOFTOUCH

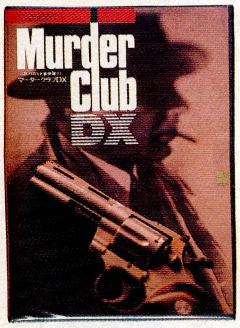
●Murder Club DX



シビアに決めて今回私は名刑事

Komura Satoshi
古村 聡

「殺人倶楽部」のX68000用DXバージョンとくれば，ゲーム内容は同じでも，その装いはまったく新しいゲーム。というわけで，今回はあの難事件に普段とはうって変わって，ハードボイルドに決めたJ.B.古村刑事が挑戦する。



X68000用　5″2HD版3枚組　7,800円
リバーヒルソフト　☎092(771)3217

奇妙な一致

ちょっとレビューに入る前に，ひとつ聞いてほしいことがある。私がこれから取り上げるゲーム「Murder Club DX」は，当然，2年ほど前X1シリーズに発売された「殺人倶楽部」の移植版である。そこで，持っている人はOh!MZ1987年2月号の「SOFTOUCH」のページを開けて見てほしい。それと今月と先月のOh!Xも用意してこの「SOFTOUCH」のコーナーを見てもらえばもっといい。

まずOh!MZ 2月号には「信長の野望・全国版」がGAME REVIEWで，そして清水和人氏が「殺人倶楽部」を2ページ使って紹介している。そして今度は新作情報のページを見てみると「口説き方教えます」,「大戦略X1」，そしてアーケードゲームからの移植として「ボスコニアン」の名前がある。なんとこれらは皆，ここ2カ月間でX68000に移植されると発表になったり，または発売されたゲームとまったく同じなのである（ででででーん！）。さらに今月のMurder Club DXのレビューの載っているのがなんと1989年2月号。うーん，もはやこうなると，偶然というより不気味としかいいようがない。ぎゃーっ!!!!

なーんてね。単なる，偶然なんだろうけど。しかし，そのX68000への移植ぐあいを見てみると，先の5作（といってもひとつまだなのもあるが）のなかでも，いまのところもっともよくできているのが，この「Murder Club DX」でしょう。信長の野望もたいへん面白いし，ボスコニアンもそれなりによくできているとは思うんだけど，Murder Club DXは，なんといっていいのか豪華絢爛というか，X68000の持つ雰囲気にピッタリとでもいうのだろうか。とにかくこのゲームの持っているなにもかもが，凄くX68000と相性がよかったといった感じなんですよね。

手がかりを捜せ

種別「殺人事件」
被害者「ビル・ロビンズ（34），ロビンズ商会社長。ロビンズ商会は自動車販売会社で，このリバティータウンでは有数の優良企業。被害者と同居中の家族は妻のジャネットのみである」
発見現場「ハウリトン・カレッジ（大学）駐車場。大学の警備員が不審な車を発見し車内を確認したところ被害者の死体を発見，警察に通報した。なお，学生が現場付近を事件のあとうろついていたらしい」
死因「かなり細い鋭利な刃物にて背後から複数回突かれ，出血多量のため死亡したものと思われる」
死亡推定時刻「8日午後1時前後」
担当刑事「ジャド・グレゴリー」

＊　＊

俺がこの報告をジャドから受け取ったのは18日。ジャドらしくもない。10日間もの間，まったく手がかりがつかめないとは。しかしどうやらジャドは，自分がリタイアする前に，この俺に事件を引き継がせるように手配したらしい。

ジャドのためにもきっちり事件を解決しなければな。フッ。

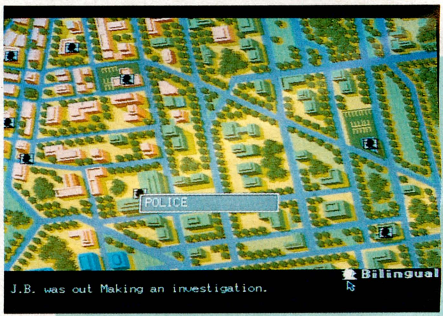
わーははは。俺ってハードボイルドが似合うなー。Gメン'75，なーんつって。これからは毎月この調子で原稿書いちゃおーかなー，はっはっは。

さて，捜査の基本は現場100回。俺は死体の発見現場へと向かった。まず，死体の発見された車を探った。が，なにも出てこない。最近の犯罪者も小賢しくなってきたものだ。次に死体の第一発見者である警備員に会い，話を聞いてみる。が，彼は本当にただ，死体を発見しただけのようだ。そうそう都合よく手がかりが見つかるわけはないと思ってはいたのだが……。

こうなったら，聞き込みしかあるまい。俺はそう考えて聞き込みに行こうと思ったが，よく考えてみるといまのところ俺の知っているところ（つまり，私の行けるところ）というのは警察，大学，ビル・ロビンズが社長をやっていたロビンズ商会，ビルの妹の嫁ぎ先ホールディング家，被害者のビル・ロビンズの家，ビルの生家エドワード・ロビンズの家，それに被害者が最後に立ち寄ったとされるパブ「ハングリーフィシャーマン」の7カ所だけなのだ。

うーむ，こんなので果たして事件解決の糸口がつかめるのだろうか？

こうして，J.B.は聞き込みを始めた。これはなかなかにたいへんな作業であったが，



マウスひとつで移動も簡単

しつこくしつこくしつこく聞き込んで，少しずつ捜査は進展していったのだ。

やっと，行ける場所も増えてきた。いま，俺は映画館の前にいる。俺はクールに映画館の窓口嬢に声をかけた。

「かのじょー，映画でも見てからお茶しない？」(こらこらこら！)

いや，違った。俺はハードボイルドだったんだ。まあ，それにしても聞き込みをしていくうちに，本当に妙なことになってくる。まず，被害者のビル・ロビンズ。こいつが実に複雑怪奇な人物である。はっきりいってしまうと，昔のテレビ時代劇の必殺仕事人あたりなら殺されてしまっても仕方のないような，ほとんど悪代官のノリの奴なんだよなあ，これが。どうやら金の絡むところ，必ず出てくる奴みたいだ。

駐車場を作るときの土地買収でも，地元の不動産屋とひともんちゃくあったらしいし，そのほかにもボロボロと出てくる。

が，しかし，それ以上に凄いのが女関係なのだ。まず，彼の秘書というのが髪の長い美人なのだが，その秘書とできていたらしい。ま，会社の社長だからそのくらいまでは想像がつく。しかし，彼の奥さんは超ド級の美人なのであった。それだけでなく，レンデル先生のところの看護婦のスージー・マクナリー（これもまた美人）とも不倫していて，パブ「ハングリーフィシャーマン」の店のピアニスト，サラ・シールズ(しつこいようだが綺麗な人なんだ，この人も）を暴行したという噂まである。

さすがに，かどわかしや囲い込みなんていうのはしてないみたいだが（時代劇じゃないっつーの)，なんかほとんど街中の美人と関わりがあるのではないかという感じである。ったく，お，おまえなんかなー，殺されて当然だーっ！（と，ついつい本音が出てしまうJ.B.古村であった)

誰が彼を殺したか？

さて，そうこうしているうちに逮捕できる奴はすべて逮捕した。家宅捜索も行った。凶器と思われるものも見つけた。

しかし，誰が彼を殺したのだろう？　アリバイなどはともかく動機に関しては皆十分すぎるくらいに持っているのだ。妻のジャネットは夫婦間の不和，秘書のシェリーや看護婦のスージーは不倫のいざこざ，どうも親子関係がうまくいってなかった節のあるエドワード，エドワード・ロビンズとつるんでいるレンデルにジェイムス・マクベイン。さらに，ビル・ロビンズはどうも他人をゆすっていたのではないか，そのた



こちらがガイ者とモメてた不動産屋

めにほかの誰かに殺されたとも考えられる。

この事件は恐ろしいことに，容疑者はいっくらでも見つかるのだがまったく真犯人が絞り込めないという，泥沼のような事態になってしまったのである。

さらにもっと凄いことに，こんな状況のときにヘンリーという謎の人物が浮かび上がってきたのだ。この，ヘンリーという男，かつてこの街にやってきて1年も経たないうちに事故死してしまったらしいのだが，なんとまぁこれが，よりによってヘンリーは現在もこの街で生きているのではないかという疑いが出てきたのだ。ヘンリーと今回のヤマとはいったいどのような関係があるのだろうか？

さらにもうひとつ。事件の手がかりを捜しているうちに，ある名簿が出てきた。この名簿には当然のように何人かの人物の名前が出ているわけだが，この名簿と事件との関わりはいったいなんなのか？　またこの名簿に絡んでいる組織（というほど大げさなものではなさそうだが）と，今回の事件との関わりは？

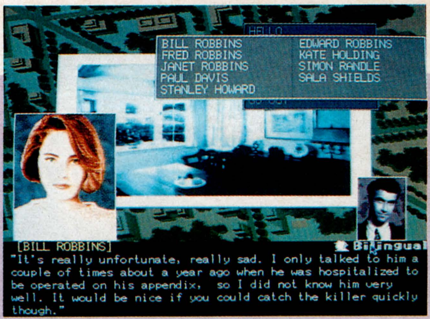
あまりにもこの事件には謎が多い。容疑者たちの動機や数々の手がかり。コロコロと供述を変える容疑者。これらは本当に，ひとつの殺人事件にすべて結び付いてくるのだろうか？　ジャド，こいつぁ，思ったより遙かに厄介なヤマみたいだぜ。

ドラマは終演を遂げた

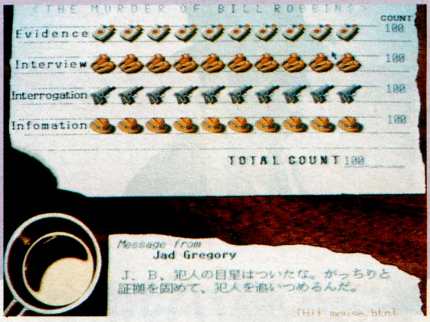
よぉ，ジャド。俺だ。ひさしぶりに見舞いに来たぜ。あのあと俺はな，ゆっくりと考え直してみたんだ。すべてはひとつの結末に向かう手がかりだとそう信じてな。やはり，事件の鍵はある人物にあった。そして，真犯人を問い詰めた結果，ついに奴は自白した。でも，俺も少しだけ安心したんだ。人を殺すような凶悪犯でも人の愛情には勝てなかったんだからな……。

*　　　　　　*

なーんて感じで，私はこのゲームを無事終わらせました。やっぱり終わったときに



どう，美人でしょこの看護婦さん



こうなりゃあとは問い詰めて自白させるだけ

はすごーく感動してしまいましたよ，うん。

さて，最初でもいったんだけど，このゲームは2年ほど前にリリースされたゲームのリメイク版なんですが，この2年の間にハードもソフトも進歩し，ほんとにすっかりその姿を変えてしまいました。このゲームはX1ではタイリングを使って普通に描いた絵なんですが，X68000では当然のようにアナログ的な色使いの中間色を使って，キャラクターや背景の画像に取り込み画像を使ってます。もう，本当にキレイ。それに，日本語/英語の表示を瞬時に切り換えられるバイリンガルモードまであって，英語モードで遊べば気分はもうすっかりアメリカしてます。

また，移動はマウスで地図上の場所を指すだけになっていて（というよりすべての操作がマウスでできるようになってるのだ。オマケに，X68000用なら当然といいきれちゃうのがこわい)，そのマップというのが上下左右にすすーっと，スムーズスクロールしてしまうのです。

マウスの使用といい（マウスの使い方もキチンと研究されている！)，スムーズスクロールといい，リバーヒルのアドベンチャーゲームは，ホントにX68000のハードをよく研究してますし，また，それを生かしたゲームの演出の腕っていうのも凄いと思います。

さて，「琥珀色の遺言」，「Murder Club DX」ときて，いったい次はなにが出てくるんでしょう？　なんにしても私は最大級に楽しみにしています。

●極道陣取り
●ザ・キングオブシカゴ





新春夢の対決 日米極道ごっこ

Ogikubo Kei
荻窪　圭

昨年は，ハルマゲドンにM of Mと，モンスターが主役の年でした。そして1989年はというと，なんとこれがやくざ屋さんからスタートするのです。それも日米競作で。というわけで，ここに夢のような(？)日米対決が始まるのです。





極道陣取り
X1/X1turbo用　5″2D版2枚組　6,800円
(2ドライブ専用)
マイクロネット　☎011(561)1370
ザ・キングオブシカゴ
X68000用　5″2HD版2枚組　12,800円
ボーステック　☎03(708)4711

「こんばんは，私が凍傷ダイモスです」。いやあ，寒いねえ。このたび引っ越しなどをして東京23区民になってしまったのだが，段ボールを開ける暇もなく，X68000とX1 turbo Zだけセッティングして原稿を書いている様は，なにやら将来に一抹の不安を抱かせるものがある。これではせっかくの新しい部屋が泣いてしまうよ。えーん，えーんとね。

なんか，最近やたら忙しくて分裂症気味で，話があっちいったりこっちいったり，ここ2，3カ月の自分勝手な現象なんだが，これを読んでいる皆さんもさぞやたいへんだろうと同情しつつ，今回も，その病に拍車をかけようとする公安の陰謀か警棒か，X68000でザ・キングオブシカゴ，X1で極道陣取りという，日米極道2本立てという暴挙がさらに私の精神状態を圧迫しつつある。

だいたい，なにが暴挙といって，この2つのゲームにはプレイヤーが，やくざかギャングかという以外にはなんの共通点もない，かたやシミュレーション，かたやシネマウェア。一方はカルーイ色もの指向，もう一方はストーリーもののシビアでシニカルなアドベンチャー風という有様が暴挙なのだ。

シネマとノベルはどう違う

極道陣取りの話は，悪いけれども少々あと回しにしておいて，最初に解決すべき問題，シネマウェアとはなんぞや，から進めることにしよう。精神状態がまともなうちにね。

シネマウェア。シネマっつうのは映画のことであるから，映画を見るように楽しむゲーム，ということ。映画のようなゲーム。ここで大戦略やドラゴンスピリットを思い浮かべる人はいないだろうが，思い浮かべたのが，マンハッタン・レクイエムだったりしたら許してあげてもいい。やはり，映画のストーリー性を取り入れたまま，ゲームとしてプレイヤーに参加させるべき要素を(つまり，プレイヤーの指先ひとつで変わるストーリーを）持たせようとしたら，どうしてもアドベンチャーみたいになってしまう。

しかし，アドベンチャーとはまったく違うゲームであったことに，私は胸をなで下ろした。シネマウェアにおいて，プレイヤーはストーリーを進める存在ではなく，ストーリーに巻き込まれる存在であったのだ。そこにはコーヒーで一服する悠長さや，無意味な選択肢のないシビアな真摯さなのだ。映画を観て，ああ，ここでこうすればあいつも助かったろうにを実践する場と思ってもいい。そういった点ではどちらかというと，アドベンチャーゲームというより，アドベンチャーゲームブックに似ているといえないこともなかろう。

では，「ソフトでハードな物語」でガンバっているノベルウェアとはどう違うのか。これはもう簡単ですね。当然，シネマウェアのほうが，ノベルウェアより字が少なくて絵が多いのだ。ディスプレイで字を読むのは結構疲れるので，やっぱ，絵が中心のほうがいいに決まってるよね。

極道2本立ての開幕

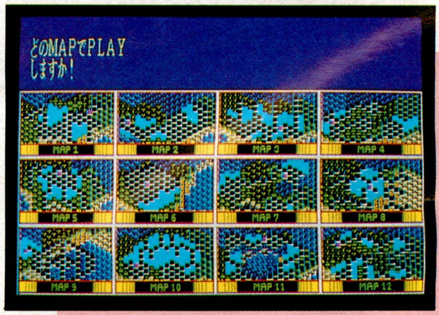
この2本立てというのは，たとえてみれば，「チンピラ」と「ゴッドファーザー」，もう少し新しいところでは，「ハワイアンドリーム」と「アンタッチャブル」といったところ。やくざ映画の菅原文太とギャング映画のアラン・ドロンといった文脈で捉えても一向にかまやしない。でも，私はといえば，鈴木清順監督の「カポネおおいに泣く」が大好き。余談だけどね。どうしてあの映画は人気がないのだろう。

このレビューが映画の2本立てと違うところは，ひとえにひとつずつ順番に観るか，一度にマルチタスクでやっちゃうか。斬った張ったの大勝負なわけ。

では，まずは例によって（どんな例なんだ？），X1さんの異色いろものシミュレーション，「極道陣取り」から。作ったのはかのマイクロネットであるからして，イロモノであるのは当たり前。グガーグガー（これも例によってディスクの音）。

親分の死に目に，なにやら遺言が残され，たかが“兄貴分”であるプレイヤーは，シマをまかされるのだった。

大戦略を遊んだことはありますか？　ある人ならわかるでしょう。画面に縮小されたマップがずらりと12個並ぶ姿。いきなり大戦略のパロディで始まるわけで，まず大笑い。死にそうな組長に呼ばれ，遺言とともに組をまかされ，さあこれからだという



極道はこうしてマップを選ぶ

ときにいきなりマップ選びとは，いったいどんな組なんだ。

とりあえず，いちばん簡単なレベルで始めよう。私はまだ駆け出しの右も左もわからない極道ざんすから。

右手のマウスだけを頼りに，四角いフィールドへ飛び出すと，そこは四角い森やら四角い街やら，四角い海，四角い野っぱらやらに覆われた変な世界。右も左もわからなくとも，優秀な部下（というより，弟分か舎弟か）が，「そこは，のっぱらですぜぇ，あにき」と教えてくれるから，大丈夫。まずは，自分の組の現状をよく見てから。

おっと，そろそろRAMディスク対応モードにした，ザ・キングオブシカゴが立ち上がったようだぜい。と，X1からマウスを抜き，X68000に差し込む（我が家にはマウスが1個しかなかったりするのだ）。

シカゴの帝王，とは

キングオブシカゴは，たいへんディスクアクセスが多いゲームであるから，ご親切にも増設RAMをお持ちの向きには，RAMディスク対応モードが用意されている。なにをしてくれるかというと，Bドライブのファイルを全部RAMディスクに転送してからゲームが始まるというただそれだけなのだが，立ち上げ時に時間を食う以外はより快適にシカゴのギャングできるってえ寸法で，こういった配慮はどんどん出てきてもらいたいと思う。

ハードディスク対応やらRAMディスク対応は，ワープロなど俗にいうビジネスアプリケーションにはよくあるけれど，ゲームだってより快適に遊べるべきなのである。ラスト・ハルマゲドンのようにディスク7枚組などというゲームは特にそうだ。うん。

なに気ないタイトルののち，シカゴの南地区を牛耳っていたアル・カポネ（それにしても変な名前）が投獄された1932年から始まる。ウォール街の大恐慌が1929年で，ルーズベルトが頑張ってニューディール政策を掲げたのが1933年だから，そのころの話だ。

ピンキー・キャラハン（こいつも変な名前）は北地区の若手有力株であるが，北地区は“じいさん”が掌握している。

主人公は，そのピンキー・キャラハン。通称ピンキーと呼ばれるが，彼の部下たちをキラーズというかどうかは知らない。近ごろの若い読者は“ピンキーとキラーズ”さえも知らない。とても，この名はギャング向きではない。やくざの親分が“みちる”という名前だったようなものだ。それでも



極道はでいりに精を出す

って，二枚目だと聞いていたのに，神経質そうないささか頼りなげな顔立ちで，こいつがボスになるのは難儀だぞと思わせるものがある。

映画でもいかにもボスになりそうな奴がボスになる話より，頼りない男がのし上がっていくうちに，だんだんとボスの顔になっていくというお話のほうが面白いからしてこれでいいのだ。ここで再び，マウスをX1に差し換える（忙しい）。

アニキはシマを広げる

組長の跡目を継いだ私は，組長の遺志も継いで，自分のシマを広げることに力を注ぐ。最初は1カ所だけであり，回りは，山やら街やらやくざに無縁な土地を除くと（街がやくざに無縁とはとても思えないが，まあ，いい），あおきくみ，たじまくみ，たんぽぽくみ，やまぐちくみ，たかみなくみなどなど実に群雄割拠，春秋戦国時代という様相でアリさんの兵隊。

黒く沈む地域は誰のシマでもないところで，隣接区域に自分のシマがあり，100万円持ってれば事務所をおったてられる。

やくざは金を儲けねばならない。そのくらいは右も左もわからないビギナーちんぴらの私でも知っている。金を儲けるには元手がいる。元手はある。使っちまわなければ，ある。それでも，建てられるのはいちばん安いのが「きゃばくら」である。名前は“ていこく”にした。きゃばくらというのはなにかというと，×××が××で××××するところである。別に伏せ字にする必要もないのだが，こうしたほうが淫靡で妖しいのよね。

金なんて放っておいても入ってくるようになると，もう立派なやくざである（きっと）。お次は，人集めだ。組員は多ければ多いほどいいし，彼らは決して裏切らない（単純なゲームだからね）。“じんいん”コマンドを選ぶのだ。「あにきー，けいきづけにでいりでもしやしょうぜ」などと，子分は脳天気である。よし，でいりだ。

行くぞぅ！　隣のシマは50人しかいない



ギャングはいきなり爆弾で勝負する



ピンキーの後ろになぜかキラーズはいない

弱小やくざだ。ケチョンケチョンに蹴散らしてやる！

ぐがーぐがー……いきなりいま流行りのベイエリアで，やくざとやくざが衝突だぁ。お！　あいつらの先頭にいる変なバイクはなんだ？　後ろにいるバズーカを持ったやつは誰だ？

おお，あっというまにやられちまったぜぃ。うーん，敵は殺し屋に暴走族まで雇っていやがった。こちとら，頭数では圧倒的だったのに。俺も雇ってやる。

じんいんコマンドだ。うーん，殺し屋がいない。もう1回。だめだ。……5回目でやっと殺し屋を雇うことができた。誰にしようかな，と。ゲゲギャ，どいつもこいつも変態みたいな名前ばかりではないか。殺し屋にまともなやつはいないのか。そういえば，ゴルゴ13だって，どうみてもまともじゃないもんなぁ。

よし，今度こそ，怒濤の大進撃だ。屋敷の大広間で，楽勝，楽勝。

シマをひとつ広げてひと安心したので，一気に日本から，タイムトリップも兼ねてウォール街大恐慌後のアメリカはシカゴへとマウスを持って出かけよう。

ギャングは気苦労が多い

あれあれ？　いつの間にかゲームが勝手に進んでしまっている。ピンキーは北地区のボスになっている。あれれあれ，髭を生やした覗き屋が銃口をこちらに向けているのはどういうわけだ。パァンッ（ズギューンというよりパァンという音なのだ），で，



ギャングは自分で相手を始末する

死んでしまって，情婦（なんか今回は危ない言葉が頻出するなあ）のローラがぶつくさいって，ピンキーの田舎の心配症のお母さんが涙を啜りながらうだうだいって（こんな母親はアメリカにだっていたのだ），結局「THE END」。

そうなのか，キングオブシカゴは放っておいても勝手にゲームが進行してしまうのか。これでは極道陣取りと交互に楽しむなどという悠長なことができないではないか。困った，困った，困ったけれども，困りついでにじっと見ていたらまたゲームが始まって，勝手にストーリーが進むさまは映画を観ているようで結構面白い。デモ画面では味わえないものだ。グラフィックもなかなかえぐいので，BGV代わりにいいのではなかろうか。そうすると，遅いのも気にならないしね。では……と，ここで終わってはレビューにならないな。

よし，気合いを入れよう。まずは北地区の覇権だが，それにはじいさんを始末するのがいちばん手っ取り早い。だからといって，むやみに殺しても，彼の部下に逆襲されるだけ。鍵はベンだ。少々年を食ったスキンヘッドの親父で，アップになった顔が不気味で楽しい。いかにも策略家だが，いざ動くとなると弱気になって漢方胃腸薬に手を伸ばしそうなタイプだ。彼を味方につけることが第一だ。そうすると，仁義なき連中も私のあとについてくる。

日本の極道と違って，杯をかわすこともなく，終身雇用制度もないアメリカのことであるから，少しでも頼りなくなると下剋上は日常茶飯事，どこで誰が裏切るかわかりゃしないから，気苦労はつきない。

たとえば，慎重派のピンキーであれば，じいさんを殺して北の街を制覇したまではよくても，ほかへ進出するのは力を蓄えてから，作戦を決めるのは参謀のベンと相談してから，というのでは，気がつくと策略家のベンに操られているような気になり，ピンキーは臆病だとの評判がたち，焦ってことを起こそうとしたときにはすでに遅く，ベンの側についた昔の部下の銃口の前にひれ伏すことになろう。

行け行けGOGO派のピンキーならば，南の奴らの銃口に狙われ，部下は怖がり，貴方をなんとか追い落とそうとし，結局は口うるさいベンを殺してしまって，彼の部下から銃弾を食らう。

気の弱いピンキーならば，じいさんを殺し損ねて，じいさんの部下にスマキにされて，湖に放りこまれるだろう。

好調なときには落とし穴がある。かといって手を抜くわけにはいかないし，情婦は（このゲーム中唯一の若い女）わがままだし。下手したら，後ろに手が回って電気イスという事態もあるのだ。おお，電気イス！

これを作った人は（もちろん，ボーステックじゃなくて，原作のアメリカの人だよ），スイもアマイも知りつくした，病んだ人に違いない。ここまで心理の襞を突かれると，もうなにもいうことはない（移植したボーステックにはいいたいことがいろいろあるけどね）。

絶対，RPGなどよりプレイヤーの性格が表れるゲームなのだ。このゲームはじっくり楽しむべきであって，キングオブシカゴを3時間で解いたよ！　などと暴言を吐くやつがいたら，スマキにしてやる。そういうゲームではないのだ。セーブもできないし，複雑にストーリーが展開するので，いちいち自分の決断を覚えてはいられないから結構毎回楽しめたりするし，長考に入ると，あっという間にストーリーが勝手に先に行ってしまうので，ぼやぼやできない。

遅いことと，アクションシーン（ドアに向かってマシンガンを撃ったり，目の前の敵を撃ったり，爆弾を投げたり）が情けないことを除けば（これは原作者のせいではないんでは？），凄く深い世界があった。どこに，情勢が悪くなると性格や顔まで卑屈になってしまうゲームがあるのだ，どうしてボスともあろうものが，ワガママな性悪女にへこへこせねばならんのだ。

ああ，人間とはなんて奥深くてあこぎなものよ。ちくしょう，暗くなっちまうぜい。ケポーンと明るい脳天気ゲームでもやって楽しく寝るとしよう。もう12時だ。

いよいよ制覇だ

明るさと暗さ，絶対値はキングオブシカゴと一緒といわれる極道陣取り。シミュレーションにありがちなコンピュータ側の長考もなく（きっと，なにも考えてないのだ，X1のやつ），余計な処理を施していないせいか，マウスの動きもスムーズである。

半分ほど制した夜更けには，キャバクラどころかパチンコ屋やソープランドも建ち（命名に困るんだよな。いくつかの候補から選ぶのだけれど，変な名前ばっかしで），金はザクザク人もうようよ。

一気に殺し屋とともに怒濤っと攻め込む。武装は最大レベル。殺し屋が5000万円要求する。いくらでも払ってやらあ，と，そんなに金があったっけ。ま，いいか。

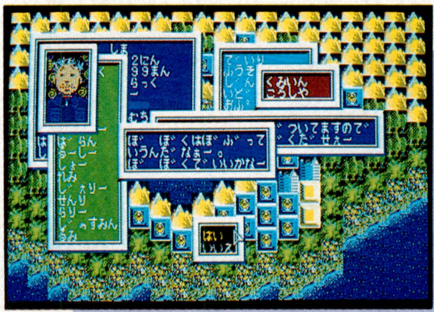
いけ！　殺し屋“ぼぶ”。どんどんどんどん。「わるいけど，おかねくれないんじゃゆるしてあげないんだな」。ぎょいえー，組がボロボロにされてしまった。態勢をまた立て直さねば。殺し屋はこわい。

で，極道は怖いのであった

極道陣取りは，なんとか勝利を簡単レベルで収め，見事，親分の遺言を実行した（なかなかシャレがきいていて気に入った）。エンディングも見た。

心残りはキングオブシカゴだ。あと一息というところで電気イスにかかって死んでしまったり，ベンに裏切られてギャングを廃業し，みんなに馬鹿にされたりと，どうもいささか頼りないピンキーの気分になりきると，ダメだとわかっていながらつい虚勢をはったり，意地になったり，短気は損気してしまうのだ。

私は悪人には向いてないらしい。とても，国会でシラを切り続けてそれが許されるからと，こりずに私利私欲に走る自民党政治家にはなれないということだ，とさ。



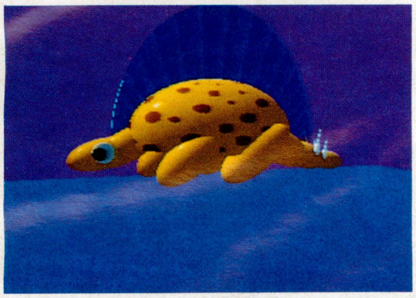
極道は慎重に殺し屋を選ぶ



ギャングはスマートに車の中で計画を練る

THE SOFTOUCH

●彩CRONE



レイトレツールにライバル登場

Tan Akihiko
丹　明彦

X68000のレイトレーシングツールにも，ようやく新しい仲間が登場しました。その名も「彩CRONE」。さて，このソフトがいったいどこまで先発隊に迫れるか，そのところを中心に美しいレイトレーシングの世界を覗いてみましょう。



X68000用　5"2HD版 2枚組　58,000円
アンス・コンサルタンツ　☎092(522)6347

X68000用のレイトレーシングソフトウェアとして独走状態にあったC-TRACEに，強力なライバルが現れた。充実したモデリング機能をひっさげて登場したレイトレーシングツール，その名はグラフィックシンセサイザー「彩CRONE」。この名前，実はPSY-CRONEときて結局は「サイクロン」と読ませるらしい。台風とはスペルが違うが，X68000の3次元グラフィック界に嵐を巻き起こせるだろうか，乞うご期待。

僕は，X68000をとりまくCGツールの発展に熱い期待をかけているひとりであると自負しているつもりだが，3次元グラフィックを扱うソフトウェアには，いまのところ「PRO-68K」の名に恥じないものは見ていない。X68000のユーザーは，単なるマニアのおもちゃに終わらない，ハイグレードなものを求めているはずなのだ。だから今回はそういった視点からこのソフトを見ていこうと思う。

彩CRONEの基本構成

彩CRONEのシステム構成は，簡単にいうと，

1)　モデリング部
2)　レンダリング部
3)　ユーティリティ群

の3つに分けられる。1)のモデリングとは簡単にいえば設計のことで，プリミティブ(基本立体)やマクロ(統括体)を使って物体の形状を定義し，色やアトリビュート（表面属性）で質感などを定義する。2)のレンダリング部はレイトレース作業を実際に行う。3)のユーティリティは，3種類の画像ファイル(X-BASIC標準のGL形式，Z'sSTAFFのZIM形式，そして彩CRONEのRGB形式）の表示・相互変換などを行う。

それでは，サンプル版を使ってみて，僕なりに感じた彩CRONEのセールスポイントや，ライバル（C-TRACE）と比べて気付いたことなどをいくつか挙げていこう。

まずは使用感のレポートから。最初はモデリングからいってみる。彩CRONEではモデリング過程にテキストエディタが一切いらない。設計に必要な処理のほとんどはコンピュータが受け持つ。ユーザーがモデリング用のプログラムに簡単なパラメータを与えていきさえすれば，あとの計算は全部やってくれる。

テキストエディタだけで考えてみると，ユーザーはこのようなツールを使う場合，次のような順序でものごとを考える。

1)　作りたいシーンを頭の中に思い描く
2)　それを図形の方程式に直す
3)　データフォーマットを思い出しながらいくつかの数値をテキストに書き込む
4)　レイトレーサを起動して，図形が思いどおりに描けているかどうかを確かめる

そこで，1)は純粋にデザイナーの感性が働く部分。2)もレイトレーシングである以上，避けては通れまい。レイトレーシングはプリミティブを組み合わせて目的の物体を作るのがオーソドックスな手法であり近年になってこの状況は変わりつつあるが(囲み記事参照)，作りたい物体をプリミティブに分解するのはレイトレーシングデザイナーの基本能力といってもいいからだ。

そして3)。まずフォーマットをきちんと守ってパラメータを書かなければならない。フォーマットを忘れてしまったときが大変だ。この手のソフトを使うとき当分はマニュアルのお世話になるものだ。それも慣れてしまえばそうは苦にならない。

いつも問題となってくるのが4)で，自分のデザインがよかったのかどうか，出来上がってみるまでわからないのだ。レイトレーシングの処理速度は上がってきてはいるが，お世辞にも「速い」とはいえない。デザインが完成するまでには，何回も描き，細かい修正まで繰り返さなくてはならない。それなのに，無用に時間のかかるデザイン行程は，いたずらにユーザーの感性の邪魔をするだけだ。どうしても，モデリングの段階でシーンが簡単に見られる工夫がほしくなる。

ともあれ，彩CRONEのモデリングは，若干不満は残るものの，このような1)～4)の要請にはほぼ応えられていると思う。一

まだ未知数のパソコン界

最近の業務用CG界では，かつては遅くて使いものにならなかったレイトレーシングの速度が実用的なレベルまで上がってきて，レイトレーシングが注目されつつある。そうなると当然研究も活発になるわけで，もっと多彩な表現も考えられている。

メタボールは，単体では普通の球だが，ほかの球が近づくとお互いに干渉・融合して変形するようになっている。これを使うと表現の自由度がかなり高まり，人の顔くらいは簡単に表現できるレベルになる。自由曲面，これはもう完全に自由な表現が可能になる。これらがレイトレーシングで扱えるようになりつつあるのだ。プログラムはパソコンでも当然記述できる。

大型機とパソコンの違いは，突き詰めていけば処理速度とメモリ容量くらいなのだから，鬼のように時間がかかることさえほんのちょっと我慢すれば，近いうちにあなたにも高等表現が思いのままになることだろう。だが，これらをパソコンのレベルでサポートしているものは，まだほとんどない。

部のパラメータについてはHELPキーで解説されるようになっていて，さらにパース・３面図をいつでも表示できるようにしてある。

さて，その操作性だが，マウス不在でキーボードしか使っていない点など，ややPC-9801版からの移植という雰囲気が強く残っている。それはそれとして，対話的環境を前提としているのは確かだ。

ただ，細かいことだが，FILE MANAGEの画面からデータ作成モードに入るとき，この彩CRONEのメニューにはなにも書いてないのにはまいった。マニュアルをひっくり返して，やっとモード切り換えがESCキーだとわかった。

これにはメニュー中に「NODE LIST」という項目を付けておけばいいのにと思うが，操作はESCキーのほうが慣れると速いので，この選択は間違ってはいない。しかし，ここでは「選択」するより，「どちらも可」とすべきだったのではないかというのが僕の私的見解だ。

まあ，これらは実際慣れてしまえばすむことで，ソフトに合わせるのはシャクに思いながらも許せる範囲ではある。ものによらず，ツールというものは意外とこういう細かいところからも，操作性について印象が大きく決まってしまうことが多い。この際どちらがいいかということが問題なのではない。どちらに決めても，それはプログラマの（勝手な）判断だ。実際に使うのはユーザーなのだから，どちらも用意しておいて，選択をユーザーに任せたほうがよかったように思う。

エディタを排除すると，必要不可欠になってくるのはデータ修正・追加の機能だが，これについても合格。C-TRACEにもモデリングソフトウエアはあったが，プリミティブを１個ずつ定義していかなくてはならない，論理演算を定義できない，マクロが作れない，セーブはできてもロードができない（編集作業の継続ができず，１回の編集で全部作り上げなくてはならない），とりわけデータを修正できない，一度定義したプリミティブは移動も回転も削除さえもできないなどの難点があったが，彩CRONEのモデラーはそのいずれもクリアしている。

ちょっぴり不満も見え隠れ

ただ，ひとつ文句をいいたい点がある。それは３面図（X，Y，Z軸方向からのシーン）と，パース（設定した構図でのシーン）の処理の不手際だ。OPT.1でパースを，OPT.2で３面図をワイヤフレームで表示する（こんな基本的なキー操作にもマニュアルが必要なのにはまいってしまう）のはいいのだが，まず思うのは，描画が遅いということ。数値演算プロセッサとFLOAT3+を使ってもとうてい許せるレベルにはない。プリミティブの数が少ないうちはいいのだが，ちょっと複雑になると，１回描くのに何分もかかる。

確かに階層化したマクロの処理は面倒だろう。特に移動や回転を伴っている場合は，計算量は増大することを考えてみれば同情の余地もある。しかし，描画にこれほど待たせるくらいなら，せめて隠線・隠面処理や，論理演算した結果まできちんと表示してほしかった。贅沢をいわせてもらうなら，ワイヤフレームモデルだけではなく，ソリッドモデル表示も望みたいところだ。

なかなかのモデリング機能

ちょっと勢いで文句をつけてしまったが，このモデラーにはいいところのほうがたくさんある。

釈CRONEのモデリングで特に感心したのは，物体の単位としてノード（NODE）を考えていることだ。C-TRACEでは，物体の単位はプリミティブ（簡単な方程式で表せる図形で，その種類は限られている。複雑な物体はプリミティブを組み合わせて作ったマクロで表現する）で，そのあと論理演算でマクロは作るものの，それがひとつの物体として扱える（扱っているような気になる）のは，せいぜい色・アトリビュートを指定するときくらいで，マクロごと動かせないのは少しばかり不便であった。

ノードとは，プリミティブとマクロを同様に扱うために便宜的に作った単位である。彩CRONEのモデリングは，ノード単位で（つまりプリミティブだけでなく，マクロまるごとでも）色・アトリビュート指定はもちろん，移動・回転も自由自在にできるという強者だ。

とにかく，マクロの管理はきちんとしている。色やアトリビュートはマクロごと指定できる。マクロはただの移動・回転だけでなく，DYNAMICモードで移動・回転（後述）でき，ノードをほかのノードの上に置いたり，そこで回転させたりできる。こうした操作は，数式的にはかなり複雑で面倒だし，テキストに記述しようと思ったらとんでもないことになるだろうが，それを感覚的なものにしている点で評価できる。

さらにデータの修正が思いのままなので，ちょっとデータを変えて描き直したいときもさして神経を使わずにすむ。この機能は，１枚絵よりも，後述するアニメーションに絶大な力を発揮する。気に入ったマクロをファイルとしてセーブできるのもgood。マクロをライブラリ化しておいて，ほかのシーンを作るときにいくらでも呼び出せる。

色を模索しているときのHELPキーは極楽。その色の球体をフルカラーで表示するので，イメージがつかみやすい。論理演算はプリミティブ名と論理演算子（*，+，−）を使って，数式の形で表す。これはシンプルながらわかりやすい。

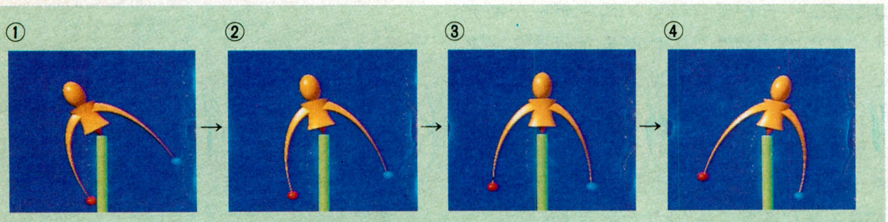
ここで，僕がこのモデラーを使ってみて引っかかったところを注意点として挙げておく。

マクロを作ると，その中心は元のプリミティブとはまったく違うところに設定されるので，手動で移動・回転を行うコマンド（MANUAL）や移動・回転をキャンセルするコマンド（ADJUST）を使うときはちょっとだけ気をつけよう。特にマクロの回転は，回転中心が自由に設定できる DYNAMICを使わないと思いどおりのことはできない。

マクロを構成するプリミティブは，論理演算で指定することになるが，そのときは，ノードリストの上のほうにあるプリミティブから順番にマクロを定義したほうが間違いが少ない。名前を付け換えるときにバグが出ることがある。これを防ぐいちばんの方法は，最初に紙の上で設計するときから綿密にしておくことだ。

おいしいアニメーション機能

速度の点を除けばまあまあ合格点をあげられる彩CRONEのモデラーがその真価を発揮するのは，アニメーションを作ろうと



これが今回，彩CRONEのアニメーション機能を使って作成したやじろべー。実際に動くところをお見せできないのが残念

いうときだ。マクロごと動かせるというたったこれだけのことで，これほど表現力が増すとは思わなかった。それに DYNAMIC MOVE/ROTの威力は絶大で，わずらわしい座標値から解放されたというのがかなりおいしかった。

あまりいい例が作れなかったが，やじろべーを作ってみた。円柱のてっぺんでゆらゆら動かしているつもりだが，これはすべてDYNAMIC ROT 1発ずつですんでいるのだ。

もちろん，レイトレで毎秒何コマものアニメなんてできるはずもない。大型機にだっておそらく無理だ。1コマずつのシーンを描画して，コマ撮り録画するのだ。最後にそれを再生すれば，立派なアニメーションだ。本気でやろうと思ったら，X68000が何台も買えるような値段のコマ撮りのできるプロ用ビデオレコーダでも買えばいいのだろうが，とりあえずは誰かビデオに詳しい人に，編集機能が優れた安いビデオデッキの存在を知らせてもらいたかったりする。手頃な値段の光ディスクが早くできることを切に願うのみだ。

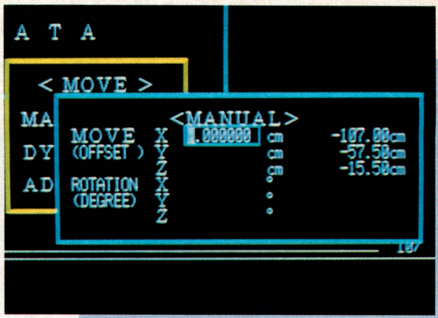
しかし，X68000のグラフィック画面はかなりアクロバティックな表示もできるので，枚数と色数さえ制限すれば，本体だけで簡単なアニメーションができる。ヒントはページ切り換え。この手法は過去のOh!Xに載っている。諸君の健闘を期待したいところだ。

それでは本番スタート

それではいよいよレンダリングにいってみよう。レンダラを起動すると，まずモデラーの出力したレイトレースファイル，続いてライティングデータを読み込む。ライティングデータは主に光源に関するデータを格納していて，レンダラで作成する。初めての場合は新規作成となる。

光源で彩CRONEはひとつ新しいことをしている。それはスポット光源だ。1方向にある角度で広がりを持った光源で，懐中電灯の光を当てたようなものである。これはいろいろと面白い表現ができそうだ。

ただ欲をいうなら，光源の設定もモデラーの仕事にしてほしかった。光源の座標や，光の方向はレイトレース画像を見てもつかめるものではない。数値だけ見てもイメージがわくはずもない（イメージがちゃんとわく人なら，モデラーも必要としない)。勢い平行光線に頼り，点光源やスポット光源を使うのが，ついついおっくうになってしまう。モデラーの3面図かパースで表示するのがベストだったと思う。



ちょっぴり簡素なメニュー画面

光源の設定が終わると本番のレイトレースだ。描画速度はC-TRACEより若干速いように思われる。

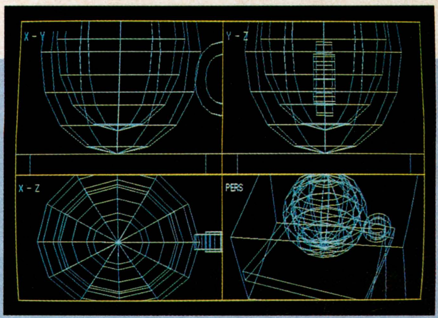
C-TRACEと比べて，レンダリングの技法が少し違うような気がする。アトリビュートの指定だが，乱反射体，鏡面体，透明体，ハーフミラーの4つに分けている。最も効率よくシーンを表現するために，簡単なモデル化を行っているのだ。C-TRACEがいくつかの物理定数の指定だけだったのとは違っているが，このこと自体は間違いではない。

タネ明かしをすると，C-TRACEでやっていた指定は，実は彩CRONEではハーフミラーに相当する。これはそれほど特殊なものではない。物理的に見ると，どんな物質だって乱反射・鏡面反射・透過・屈折などをしている（実際に人の目に見えているかどうかはこの際別として）と見なせば，C-TRACEのやり方もやはり正しい。で，結局どちらを取るかは，データ構造の統一性か，処理の速さかの選択の問題になると思う。

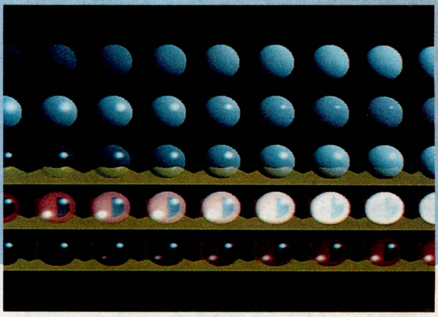
さて，肝心のレイトレーサとしての実力であるが，やや力不足の感は免れない。まず各種パラメータの指定が不足していて，結果として細やかな表現ができない。

また環境光線強度（C-TRACEでいうならアンビエント，周辺光）についても無色に固定である。確かに周辺光は無色のことが普通なのだが，そう決めつける必要はどこにもない。それから，屈折率も1.2に固定してある。マニュアルを見ると，「RAY-TREKIIとリンク時のみ設定」とある。RAY-TREKII を使うときにだけ，1.2以外の値にできるのだ。もちろん屈折はちゃんとするから，それほどは困らない。しかし，ほかにもマッピング範囲もマッピングモードも指定できない。これらは決して独自の感覚で勝手に決めてしまってはいけない。使うのはユーザーなのだ。

さらに残念なのは，バンプマッピング(表面の凸凹・ザラザラ感の表現）はできないということだろう。広告にはバンプマッピングを使った紙風船も見えるが，あれを描くにはPC-9801とRAY-TREKIIが必要だ。



3Dモデルの移動・回転は自由自在


サンプルで付いている条件別データ

ライバルに負けるな

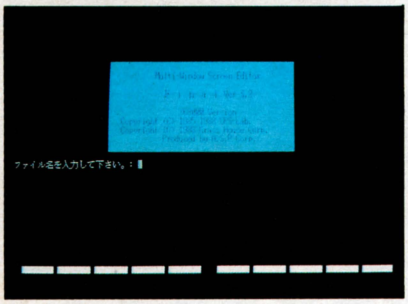
彩CRONE についてまとめてみると，モデラーは速度的に少し不満だが高性能で魅力的，一方レンダラは速度はまともだがレイトレーサとしての機能は基本的なものにとどまっている。といったところだろう。

レイトレでは後発のソフトということで，個人的にはC-TRACEを上回るようなレイトレーシング機能を期待していた。その意味では，彩CRONEよりもC-TRACEのほうがレンダラの機能はずっと強力で僕としてはかなり不満が残る。マクロの情報だって，レンダラが直接扱っているのではなく，モデラーがレンダラに渡しているデータのフォーマットは平凡な形式だった。

とはいえ，モデラーが強力なのは事実。はっきりいってレイトレーシングは初心者には難しいものだ。だから，今回私がここで紹介したようなグラフィックを自分でも作ってみたいという人には彩CRONEのほうがいいかもしれない。

ところで，彩CRONEを開発したアンス・コンサルタンツは，PC-9801を使った大がかりなレイトレシステムも出している。その名も，プレゼンテーションウエポン「天彩児」。締めて250万円ナリ。しかし，どうしてこうネーミングがハデなのばかりなんだろう。

●Final Ver.3.2



さらに効率アップ 新エディタ登場

Noda Naoki
野田 直樹

それぞれのユーザーが独自に持っている開発環境をより効率のいいものにしてくれるエディタ。そこに秘められた豊富な機能とは，どのようなスタイルのものなのでしょうか。まずはその概要を中心にスポットを当ててみることにしましょう。



X68000用　5"2HD版　38,000円
エー・エス・ピー　☎03(767)1451

突然ですが，私はいまFinalを使って原稿を書いているところです。日本語FPはASKです。ドキュメントモードというのを使うと指定したところで自動的に改行してくれたりして，思わずワープロを使っているような気になります。

このFinal。ずいぶんと前にやっと本格的なエディタが出ますよ，と紹介されたまま誰にも忘れ去られ，本家本元のPC-98版ではいつのまにかVer4.0にバージョンアップし，とうとうX68000版は忘れ去られたのか？

と思わせておいて，やっと，苦労のあとも濃く，発売されることになりました。ここまでずいぶんと時間がかかりましたが，さすが評判のエディタということもあって，カーソル移動は速いし，スクロールはもうヒューヒューと新幹線だし，カーソルの移動速度は変えられるわ，キーリピート速度も変えられるわで結構至れり尽くせりです。

今回はまだ完成直前のサンプル版を使っていることもあって，Finalとはいったいどのような性格と機能を備えたエディタなのかを中心に紹介していくことにしましょう。

まずは概要から

Finalはどこからどう紹介すればいいか迷うほどたくさんのオプションスイッチやモードを持っています。だから，使う人は，目的に応じた立ち上げ用バッチファイルをいくつも用意することになるでしょう。

モードは3つあります。ひとつ目はいうまでもない通常のエディタモード。2番目は冒頭でも紹介したドキュメントモード。このモードですと，任意の桁数で自動改行をしてくれるので（しかも改行といっても，改行コードではなく特殊なソフトウェアキャリッジリターン。なんのことはない^¥なのだが）が書き込まれるだけなので，文書自体が改行だらけになるわけではありません。

3つ目はバイナリエディットモード。画面にダンプが出て，それを直接修正してしまうという実に便利なモードなわけです。いちいち正式な手順を踏むまでもないファイルのちょっとした修正はよくあることですから，そういったときはバイナリモードで，ちょいと直してやればいいのです。

モードが決まったら，次はオプションです。たいていは起動後もメニューやコマンドによって変えられますが，カーソルの移動スピードなどのような初期設定のみのオプションもあります。たとえば，私は「－U」オプションを使って，G-RAMを作業用エリアにして高速化を図っています。

オプションの話をついでにしておきましょう。起動後は，SHIFT＋F1で設定の変更メニューがプルダウンします（といっても，マウス対応ではない）。そこにはBOXカットペースト（四角く範囲指定をするモード。通常は文字列カットペーストである），フリーカーソル（画面上を自由にカーソルが動く。通常は入力された範囲でしかカーソルは動かないが，それでも，上下移動の場合はスタート時のカラム位置を覚えており，便利），改行文字の表示のON/OFF，EOFの表示のON/OFF，TAB文字の表示のON/OFF（これらはお馴染みだよね）もあります。

このほかにも画面の状態を設定するものとしては，アンダーライン（カーソルのある行にアンダーラインを引いてくれる），行番号（カーソル位置が何行目かすぐわかる），カラムゲージ，オートインデントなどなどとても便利です。画面分割形状の変更もできます（縦割りか横割りか）。

初心者からマニアまで

Finalのおいしいところは，まだ使い方をマスターしていない人はそれなりに，ばしばしプログラミングするブラインドタッチ派という人にもそれなりに使える点でしょう。

初心者の方は，面倒なコマンド体系を覚えなくても，ファンクションキーにはメニューが収められているので，検索・置換やジャンプ，ウィンドウのオープン・クローズやエディタの各種終了，子プロセスの起動などができます。ヘルプメニューも，よくあるHELPキーを押したときにだけ登場するヘルプメッセージのそれとは違い，画面の上7行がヘルプ用になり，ヘルプメッセージが常駐してしまうのです。このあたりが私がFinalを好む理由のひとつなのです。ヘルプウィンドウを出したままファイルのエディットができるのですからこれは楽ちん。

このように，初心者やエディタの機能をみんな覚えるのはいやだけど使いこなした



常駐可能なヘルプメッセージ

いという人にも親切なのです。

中級以上のレベルであると自負している方は，コントロールキーを駆使して，自在にファイルを作りましょう。それがもしCのプログラムだったりすると，Finalを使えばEDでは絶対に望めなかった環境が味わえるはずです。このFinalは，Cでコーディングする人たちを相手に商売しているはずなのですから。

たとえば，オプションで「−IC」とすると，C言語用のオートインデントをしてくれます。「{」で改行すると，次の行の頭が「{」の位置よりさらにTAB分だけ字下げされるのです。「}」のときはその逆。また，プルダウンメニュー4にはCコンパイラを起動するコマンドが入っています。そのうえ，タグジャンプ機能がデバッグにひと役かってくれます。

まず，Cコンパイラのメッセージの出力を適当なファイルに書き出します。次にそのファイルを読み込んで，捜したいエラーのある行でタグジャンプを実行すると，ソースファイルのその行へジャンプしてくれるのです。ま，これはED.Xをはじめとして，X68000用のスクリーンエディタでは当たり前の機能ではありますが，Cを使う際には便利な機能といえるでしょう。

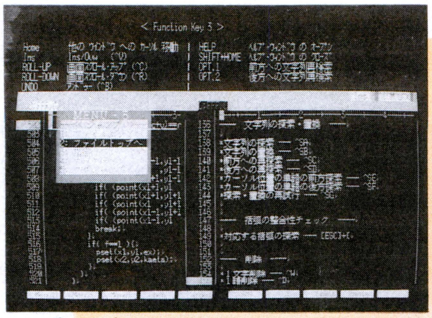
また，「CCREF」というユーティリティも付いています。これは，関数定義行から関数を呼び出している行やらマクロ定義関係の行を洗い出したりしてくれるもので，これにひと肌脱いでもらえば，Finalで関数定義行へのジャンプコマンドを使うだけで，関数定義行へとジャンプしてくれるのです。このように，プログラミングするユーザーを手助けするエディタというのが，Finalの正式な顔でしょう。

ただ，Finalは150Kバイト以上あるので，EDに比べて機動力で若干劣っているようです。ま，これは各人の好みによって判断が変わってくるところでしょう。

使い方の基礎知識

以上のようにFinalは非常に使い勝手のいいエディタです。では，基本的な使い方にも簡単に触れておきましょう。

Finalは，ほとんどの場合バッチファイルで起動することになると思います。ファイル名を指定しないで起動すると，当然の如くファイル名を聞いてくるので，そこで，開きたいファイルのあるドライブを「B:」というように入力してリターンキーを押すと，ウィンドウが開いて，ファイルがびっしりと並びます。そこで，カーソルを合わ



シンプルなプルダウンメニュー

せてリターンキーでOK。

すると，オプションで指定した，あるいはデフォルトの編集画面になります。ここでおもむろに，まずヘルプ画面を上7行に常駐させましょう。PC-98版だと画面が80×25行であるから7行も取られるとかなりつらいものがありますが，X68000では96×32行だからそう気にはなりません。なお，ヘルプ画面はヘルプファイルがカレントディレクトリにないと開かないので注意が必要です。

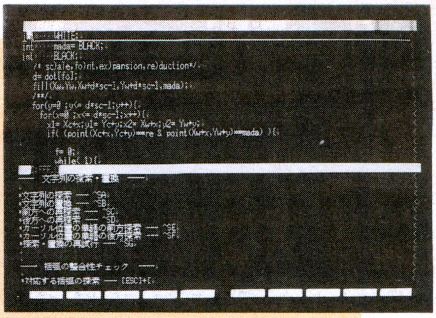
では，編集を始めます。最初に述べたように，CTRLキーを使います。

こうして使っているうちに，「ヤバイ，間違って消してはいけないものを消してしまった」となります。すると，UNDOキーの出番です。ちゃんとUNDOしてくれます。ついでに，X68000版ではOPT.1とOPT.2キーにも機能を割り振ってくれています。それは前方再検索と後方再検索です。さらに，XF5キーにも短縮入力（ワープロでいう，短文登録みたいなことができる）が割り振ってあるので，WINDEXとまではいかなくとも，結構X68000的な仕上がりを見せてくれます。

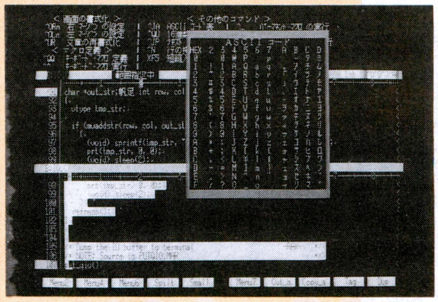
慣れないとCTRLキーの使い過ぎで左手小指がひきつるかもしれませんが，それだけ快適な機能が揃っているわけです。

私のように文章も書く人には，前述したドキュメントモードもいいですが，右マージンを設定して（この原稿は19字詰めだから38），「^UR」を押せば，綺麗に19字詰めで文章を整理してくれます。

Finalの編集機能の大きな特徴として，検索機能の豊富さにも触れておきましょう。なんと，正規表現が使えるのです。UNIXに精通している人しか知らないでしょうが（もっとも，私は卒論のプログラムをUNIX上で作ったが正規表現は知らなかった），これはとても便利なものです。検索時に，Humanでディレクトリを見るとき便利なワイルドカードのずっと高級なやつが使えるのです。長いプログラムになると，絶対，この機能



縦横分割や入れ換えも自在



作業中にASCIIコード表も呼び出せる

がほしくなるものです。

そんなこんなで，編集が終われば，コンパイルして，ころころとやって，終わればセーブして終了。ああ，こんなことしてこんなことがしたい，と思ったら，即，パーマネントマクロファイルを作りましょう。

EDではひとつのキーボードマクロしかありませんでした。FinalではEDのようなキーボードマクロのほかに，パーマネントマクロといって，あらかじめマクロを登録しておくことができるのです。

最後にコンフィギュレーション

マクロの話ついでに，コンフィギュレーションの話をしておきましょう。Finalでは，そのほかの優秀なエディタと同様，好きなようにキーを割り振ることができます。たとえば，CUSTOMというディレクトリには，PMATE用コンフィギュレーションファイルとヘルプファイル，WordMaster用のコンフィギュレーションファイルとヘルプファイルなどがついてきます。

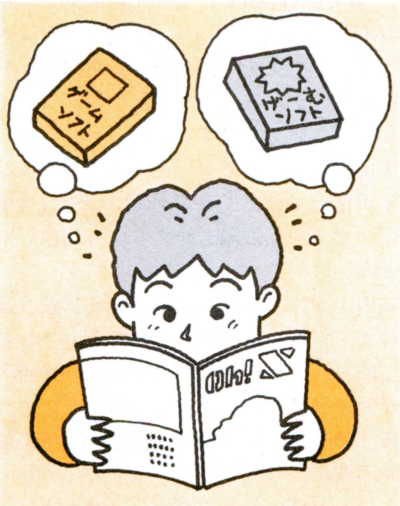
専用のコンフィギュレート用ユーティリティが付いてきますので，EMACS風とかEDオリジナル拡張編とかするのも面白いでしょう。

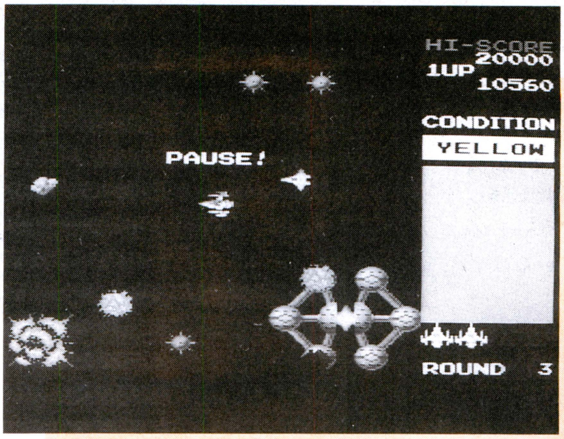
このように，このFinalは，使えば使うほど骨までしゃぶれるエディタであり，これまでEDだけに頼ってきた環境に不満を持つあなたの強い味方となってくれそうです。今後，X68000におけるこのような環境整備週間の波が，いつやってくるものなのか，のんびりと待つのもこれまた一興です。

THE SOFTOUCH

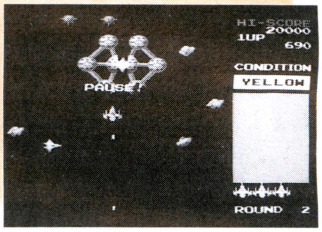
SOFTOUCH PRO-68K

サンダーブレード
アフターバーナー
第4のユニット2
アークス
Might&Magic2
ウルティマI
美少女写真館スペシャル・ダブルヴィジョン
カインドゥ・ギャルズ
Z'sSTAFF PRO-68K [Ver.2]
New PrintShop PRO-68K「GRAPHIC LIBRARY VOL.1/VOL.2」





いやー，お懐かしい。機種は変われど，やっぱりボスコニアンはボスコニアンなのでした。でも，少し英語が達者で爆発シーンが妙にリアルになったのかな，このX68000版は



「話題のソフトウェア」のコーナーでは，ゲーム新作情報を簡単にご紹介しましたので，ここでは，X68000のツール関係のソフト情報中心にお届けします。

まず最初は，ワープロソフトWORD PRO-68Kと，そしてMIDI対応になってカムバックのMUSIC PRO-68K［MIDI］のお話。このWORD PRO-68Kの概要に関してはまだ詳しいことはわかっていませんが，どちらかといえばワープロに図形処理機能が合体したDTP指向のワープロソフトとなりそう。一方のMUSIC PRO-68K［MIDI］は，その名のとおり現在発売されている楽譜ワープロ，MUSIC PRO-68KのMIDI対応版で，先に発売されたPC-98版のMUSIC PRO-98［MIDI］の持つ機能にX68000のオリジナル機能がプラスされての登場ということです。どちらも発売時期はまだ未定ですが，MUSIC PRO-68K［MIDI］，WORD PRO-68Kといった順で春先から初夏にかけて発売の予定とか。さて，ワープロはいったいどのようなスタイルになるのでしょうか。

次に発売されたばかりのOS-9/X68000には，**C&プロフェッショナル・パッケージ**や**プログラマーズ・ツール・キット**など4本のツールがマイクロウェア・ジャパンから発売されます。これらOS-9関係のアプリケーションについては，概要はこのコーナーで，そして詳細については連載の「OS-9/X68000入門」のほうで順次レポートしていく予定ですので，OS-9ユーザーの方も期待していてくださいね。

X68000ソフト&ツールズ

☆……12月25日現在発売中　★……近日発売予定

★サンダーブレード

シューティングゲーム「サンダーブレード」が，シャープから発売される。このサンダーブレード，最近ではメガドライブに発売されてお馴染みとなっているが，もともとはセガのアーケードゲームで，アフターバーナーに続くセガの自信作として発表されたものである。このゲームの構成は，スペースハリアーのような3D画面モードと，上方から見た立体的な縦スクロールモードがあり，その2つのモードをヘリコプターを操り，市街地や岩地をくぐり抜けながら暴れ回るというスリルあふれるゲームだ。特に障害物の間をすり抜けるスピード感は爽快。電波新聞から発売されるアフターバーナーとともに発売の待ちどおしい1本といえよう。このサンダーブレードは，パックマニアに続いて春ごろの登場となりそう。

X68000用　5"2HD版　価格未定
シャープ　☎03(260)1161

★アフターバーナー

電波新聞社からは，あの「アフターバーナー」が間もなく登場する。このアフターバーナーはセガの体感ゲーム第5弾として開発されたもので，一昨年，アーケードでの人気を独り占めしていたのはまだ記憶に新しい。とにかく，空母から発進したF-14トムキャットによるドッグファイトは圧巻。煙を吐き飛ぶミサイル，左右旋回で地上が360度回転するなど，そのスピード感はこれまでのシューティングゲームのイメージを一新してしまったともいえる名作。あの興奮が間もなくX68000でも味わえる。

X68000用　5"2HD版　価格未定
電波新聞社　☎03(445)6111

☆第4のユニット2

ユニークなゲーム構成で好評の「第4のユニット」，その続編がX68000にも登場だ。メッセージ数は前作の2倍，そしてさらにパワーアップした戦闘モードなど，より強力な仕上がりとなっている。今回のストーリーは，BS計画によって作られ



第4のユニット2

た4番目のユニット「ブロンウィン」の前に強力なライバルが現れる。それは，敵の作った5番目のユニット「ダルジィ」であった。いま，巨大な敵WWWFを前に，越中博士が，セスが，そしてブロンウィンが立ち上がる。ここにハードアクション・ハイパーバトルの世界が，いま，再び始動する。

X68000用　5″2HD版2枚組　7,600円
データウエスト　☎06(968)2792

★アークス

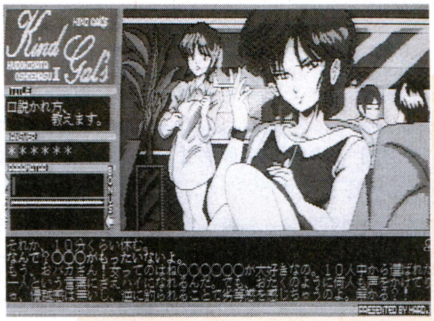
X1turboで評判だったウルフチーム初のRPG「アークス」が，X68000にも発売されることとなった。このゲームは，ウルフチームでは初めてのX68000用ゲームとなる作品で，他機種用のものとは違い，戦闘システムや移動システムに新しく手を加えられての登場である。物語は，人間が精霊たちの使う力をいつしか覚えて使えるようになり，それを魔法と呼び始めたころ。人間たちは己の欲のため，よりよい土地を奪い取り，先住民を追い立てるためその力「呪文」を使うようになってしまった。そうして，そのあまりの人間の無軌道ぶりを見かねた精霊たちは，アルカサスの国を見捨てて出て行ってしまった。その後，精霊たちを失った王国では，至るところで大地は荒れ，人の心はすさみ，この国にかつてない異変が起きようとしていた。そのときジェダ・チャフを含む6人の旅人たちが，それぞれの目的を持ってアルカサスを目指していた。こうして，この6人は運命的な出会いを遂げ，次第にさらに大きな運命の渦のなかへと巻き込まれていく。

X68000用　5″2HD版　価格未定
ウルフチーム　☎03(269)8650

★Might&Magic2

弟子のグウィンドンの話によれば，神秘の人コーラックは突如として我々の住む世界から謎の失踪を遂げた。彼はそれまでの間，異常な行動が続いていた。そして，平和の地クロンにシェルテムという危険な異星人が現れること，そのシェルテムの率いる軍勢はこのクロンの地を支配しようとするであろうこと，そして聖なる騎士が現れてその力でクロンがほかの世界と同盟を結ばなければクロンは滅亡するであろうことなどを予言していたという。いったい謎の異星人シェルテムとは何者なのか。彼はこのクロンでなにをしようというのか？　こうしているうちにコーラックの弟子の話は次第に現実となり，その情け容赦のない侵略に突如現れた聖なる騎士が立ち向かっていくこととなるのだが，その行く手に待ち受けるシェルテムとはいったい……。こうして展開されるM&M2は，200種類以上のモンスター，90種類以上の呪文，200種以上のアイテムなど前作を遥かに上回る規模で登場する。なお，X1/X1turbo版もそれぞれの機種の専用版として発売される。

X68000用　5″2HD版　9,800円
X1/X1turbo用　5″2D版　9,800円
スタークラフト　☎03(988)2988



カインドゥ・ギャルズ

★ウルティマⅠ

反乱は起きた。永年に渡って慈悲深い偉大な王ロード・ブリティッシュのもと繁栄してきたブリタニアの国に。悪の魔道士モンデインが密かに魔法の力を身に付け，長い間この国を征服する機会を狙っていた邪悪なものたちとともに，この国の人々をその悪の意志のもとに支配したのである。こうして，ブリタニアに暗黒の時代は始まった。そしてモンデインの造り上げようとする「賢者の石」を破壊し平和を取り戻すべく，勇者の旅が始まる。あの古典的なロールプレイングゲーム，ウルティマⅠ「The First Age of Darkness」がリメイクされて戻って来た。今回のX68000版はグラフィックの描き直し，メッセージの変更などあらゆる点に変更が加えられての登場だ。このウルティマⅠに続いてⅡ，Ⅲも続いて登場の予定。また，X1/X1turbo用も発売が決定している。

X68000用　5″2HD版　7,800円
X1/X1turbo用　5″2D版　7,800円
ポニーキャニオン　☎03(221)3161

☆美少女写真館スペシャル・ダブルヴィジョン

このゲームは他機種では，美少女写真館1「スタジオ・カット編」，2「ムービングスクール編」という2本のソフトだったものを，今回のX68000シリーズ用ではひとつにまとめられている。ゲームは被写体の女の子をさまざまな場所で，一眼レフ，プロ用カメラなど3つのなかから選んで撮影するのだが，露出やシャッタースピードをうまく合わさないと，女の子の写真がきれいに画面上に表示されないというその筋のゲーム。

X68000用　5″2HD版　5,800円
ハード　☎03(837)1893


美少女写真館スペシャル・ダブルヴィジョン

☆カインドゥ・ギャルズ

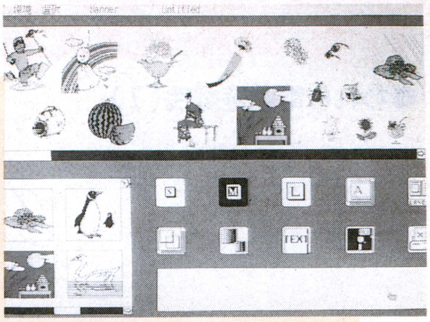
ハードから発売中の「口説き方教えます」のパート2，カインドゥ・ギャルズが登場。ゲーム内容はほぼ前作と同じで，男女2人の会話のなかで空いているセリフの部分を埋めていき，正解であれば，女の子がナンパできるというゲーム。そして5人の女の子をナンパできれば無事エンディングを迎えられる。とってもその筋なゲームなのでその筋の方はどうぞ。

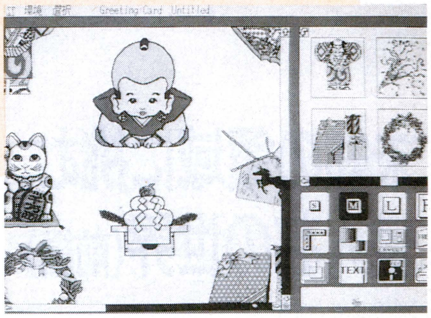
X68000用　5″2HD版2枚組　6,800円
ハード　☎03(837)1893

★Z'sSTAFF PRO-68K[Ver.2]

グラフィックツールとして定評のある，Z'sSTAFF PRO-68Kが今回バージョンアップされて新登場した。このバージョンアップ版は，従来の豊富な機能に加えて，JIS第1水準の明朝・ゴシック体のアウトラインフォントを搭載したことと，入出力機器のサポートが豊富になったことである。入力機としては，マウスのほかに各種イメージスキャナにも対応。シャープのJX-200やNECのIN-501，エプソンのGT-3000など4社8タイプのイメージスキャナに対応，イメージユニットを介してのビデオカメラによる画像入力なども可能となっている。また，出力機としては4社23タイプのプリンタに対応し，A4サイズ縦やハガキサイズのハードコピーも可能となっている。今回のバージョンアップに関して，ユーザー登録をしているユーザーには詳細の書かれた案内状を送付，または無償バージョンアップシールの付いたZ'sSTAFF PRO-68Kを購入したユーザーには，発売元のツァイトがバージョンアップサービスを行ってくれる。

X68000用　5″2HD版5枚組　58,000円
ツァイト　☎03(299)0461


GRAPHIC LIBRALY VOL.1


GRAPHIC LIBRALY. VOL.2

☆NEW PrintShop PRO-68K「GRAPHIC LIBRARY VOL.1/VOL.2」

NEW PrintShop PRO-68K用のグラフィックライブラリが2本，シャープより同時に発売された。このライブラリには，VOL.1には毛筆体の暑中・残暑見舞い用の文書のほかに，夏の季節に合わせた花火や風鈴，祭り，七夕などのグラフィックデータがSサイズ12種，M20種，L10種，フルパネル5種，レターヘッド5種，ボーダー10種が収められている。一方のVOL.2には，年賀状を中心とした毛筆体文書や干支，宝船のほか，12月用のサンタクロースや節分，ひな祭り，端午の節句などのグラフィックデータがSサイズ30種，M30種，L15種，フルパネル7種，レターヘッド9種，ボーダー10種が収録されている。これらのライブラリ集では本体ソフトのなかには比較的少なかったM/Lサイズのサンプルが大幅に増やされているのが特徴で，また，付属のデータコンバータによりZ'sSTAFF や X-BASICからの使用が可能となっている。

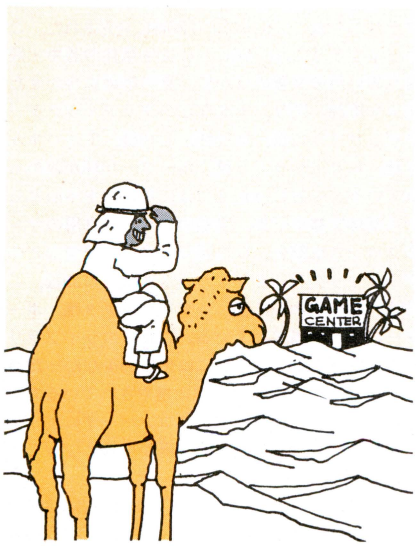
X68000用　5″2HD版　各8,800円
シャープ　☎03(260)1161

●われら電脳遊戯民(7)

未知の領域に挑む職人芸の世界(前編)

Kuramochi Ryouichi
倉持 亮一

ゲームの歴史を次々と塗り替えてきたデザイナーたち。しかし，時代はいつまでも彼らに依存しているわけではない。次世代へのカギを握るのはいったい誰なのか。壮大なテーマでお届けする今回のわれら電脳遊戯民，まずはその前編です。



この連載の第4回の「ゲームとアイドルの相関関係を探る」で，私は冒頭にドラゴンクエストIIIの話を紹介した。今回は別にその続きというわけではないが，スーパーマリオ3の話から始めようと思う。「なんやこいつ，またファミコンソフトの話かいな」などと関西バージョンで突っ込みをかけてくる方も，とにかく今回の話のマクラとして聞いていただきたい。

とにかくスーパーマリオ3

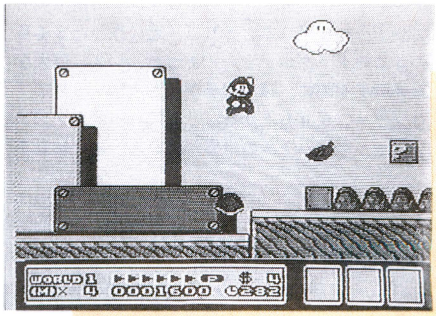
昨年10月に発売された任天堂のファミコンソフト「スーパーマリオブラザーズ3」は，ファミコン自体の人気が衰えてきたせいかそれほど話題にはならなかったけれど，これまでのシリーズのなかで最も完成度の高いゲームとして雑誌などでは昨年秋ごろさかんに取り上げられていた。確かにこの3は，隠れキャラや難度が少々鼻についた初代スーパーマリオ（1985年9月発売）に比べてかなり洗練され，プレイヤーを驚かす仕掛けやアイデアなどが豊富に盛り込まれていたのだ。

たとえば，このゲームのなかに隠れブロックというものがある。普通はなにもない画面なのだが，ある場所でジャンプするとブロックが突如画面上に出現するというアレである。これが，ある場所では上からは降りられるが，下からはジャンプできない一方通行のトラップになっていたりする。単なる隠れキャラの存在だって，このように使えばあっという間にトラップとして変身するわけである。

また，前作の2までは火の棒が一端を中心に回転するファイアーバーだったものが，今回の3ではダメージを与えない核(コア)を中心として一定距離をおきながらクルクル回る「クッキー」という敵キャラになっている。これによって，核の部分とクッキー本体の回転している部分の間をタイミングよく，くぐり抜けたり，核の部分にジャンプしてそこで待機したりというように，プレイするバリエーションも大幅に広がっている。

このほかにも，静止と回転を繰り返し，いざマリオが飛び乗ろうとすると，あらぬ方向に弾き飛ばされる回転板など，とにかくプレイヤーを楽しませるサービス精神が旺盛なのだ。

どこかのファミコン誌では，このゲームを「職人芸といっていいのではないか」と書いてあったが，確かにトコトンまでプレイヤーを驚かせ，楽しませようとする姿勢にはゲームデザイナーやプログラマのセンスよりも，このトータルな意味での“職人芸”による部分が大きいように思う。



©任天堂

ゲームの職人(達人ではないぞ)

考えてみれば，広く支持されてきたゲームのなかには，この職人芸と呼ばれる領域で完成されたものが多い。たとえば，ドラクエシリーズである（また出てしまった）。このソフトのゲームシステムそのものは，結局は「ウルティマ」や「ウィザードリィ」の美味しいところをマネたもので，独創性に富んでいるとは決していいがたいものだが，プレイヤーを引き込むシナリオや戦闘システムのよさなど，プレイヤーをドラクエの世界に引きずり込む「ワクワクさせる作り方」という点においては，圧倒的に卓越したものがあるからこそ高い人気を得たのである。

そういう点ではイースシリーズも同様だろう。ゲームのスタイルやストーリーはオーソドックスなものであるが，プレイヤーが心地よくプレイできるといったバランスにおいては，もう完全に職人芸によるものであるといっていい。

もちろん，こういった職人芸は日本に限ったものではない。かの「ウィザードリィ」を作り上げた，ロバート・ウッドヘッドも当然そのひとりであろう。いまでは「RPGにおけるポイントはシナリオだ」といわれるようになっているが，このウィザードリィには完成されたシナリオというものはほとんどなかった。それでも，このゲームが日米両国において数多くのプレイヤーを虜にし，現在でもシナリオ4が発売されるという人気を誇っている背景には，当時は斬新だった戦闘システムや，転職システム，また滅多に見つけ出せないアイテムなど，このゲームだけが持つ魅力的な輝きを秘めていたからなのだ。だから「MURAMASA BLADE!」や「GARB OF LORDS」を探し求めて，いったい何人の人たちが夜明けをウィザードリィとともに迎えたことか。

ひと言でいって，結局は職人芸は各個人

の持つ才能に依存するところが多い。ロバート・ウッドヘッドなどは，プレイヤーをのめり込ませる才能といった意味では，あの時代においていわば頂点にいた存在なのである。

こういう職人芸は，RPG以外のゲームにも存在する。業務用ビデオゲームでは3作目（あれ，今回はずいぶんと3に縁がありますな）に突入した「グラディウスシリーズ」。年ごとにシューティングゲームはシロウト集団を排除するかのごとくレベルアップを続け，このグラディウスもご多分に漏れずといった感があるが，それでも数あるシューティングゲームのなかにあって頭ひとつ抜きん出た人気を誇っている。

実はこのゲーム，最初は死んで当たり前ではないかという攻撃を仕掛けてくる，一見とんでもないゲームのように思えるが，その攻撃網をちゃんとくぐり抜けられるような「抜け道」が巧みに用意されていて，決して支離滅裂な無差別攻撃だけではない。それを見つけ出すことがこのゲームの面白さの核となっているともいえよう。

だからちょいとゲームに手慣れている人であれば，何度かやっているうちにギリギリですり抜けられる快感が味わえるのである。こういう作り方を考え，実行できるというのがここでいう職人芸の域であり，また完成されたかたちであると私は思っている。

ここに「アーティスト」という外来語がある。最近は本来の「芸術家」という意味よりも，「創造的な仕事をする人」という意味で使われている。なにしろアイドル歌手でも，アーティスト志向のタレントが登場したりするご時勢である。ゲーム制作においても当然，独創的な仕事であるからにして，ゲームデザイナーをアーティストと呼ぶことには支障がないと思う。

しかし，最近はゲーム作りの職人，アーティストに対する言葉でいえば「アーティザン（artisan：職人）」の手による「アーティザナル（artisanal：職人的）」なゲームに人気が集中しているのではないのだろうか。そしてこのアーティザナルとは，ゲームだけではなく，最近の娯楽文化全般にも影響を及ぼし始めているのだ。

娯楽のなかのアーティザン

娯楽文化全般といっても，別に，いまパチンコ屋さんが面白いとか，危ないお風呂屋さんの営業展開がどうのというあちら向けの話ではなく，話は皆さんお馴染みの「ルパン三世・カリオストロの城」の映画の話題から始まるのである。

この映画は，「風の谷のナウシカ」や「となりのトトロ」といったアニメ映画でお馴染みの宮崎駿氏が，単独で脚本，監督を担当した初の劇場用長編アニメである。この「カリオストロの城」は，アニメファンの間ではずいぶんと評価が高い作品だったのだが，宮崎氏自身はこの作品を称して「大棚ざらえ的作品」だと述べている。

この映画のなかで展開される世界を，いくつかのキーワードで表すとすれば，「塔に幽閉されたお姫様」，「王家の血の秘密」，「陰謀を企む家臣」，「時計台での決闘」，「王家に伝わる謎の遺言」というどれもヨーロッパの昔の推理・冒険小説でよく使い古されたネタであって，決して目新しいものではない。

しかし，それでも結果として高い評価を受けているのは，キャラクターの動きや表情，セリフなど各部分部分に宮崎氏の動画家としての職人芸が生かされていて，観客に「ワクワクする」，「面白い」といった評価を与える作品に仕上がっているからなのである。

こういう職人芸ではジョージ・ルーカスも似たようなものを持っている。かつて大ヒットした「スターウォーズ」も，そのテーマやストーリーだけを考えると，それほどの斬新さはない。

上映当時話題となった特撮も，ミニチュアのセットを使ったシーンだけをとってみると，最初の作品では8分くらいのものだ。それでもこの映画が映画史上に残る話題作となってしまった背景には，観客を楽しませるための職人芸があればこそなのだ。

彼の観客を楽しませる職人芸は，あの魅力的なキャラクターたちにも現れている。たとえば，全身真っ黒装束のゴキブリみたいなダースベイダーも，ヘルメットに丸みを付けて，イヤミになりがちなキャラクターの威圧感を和らげる配慮がなされていたりする。

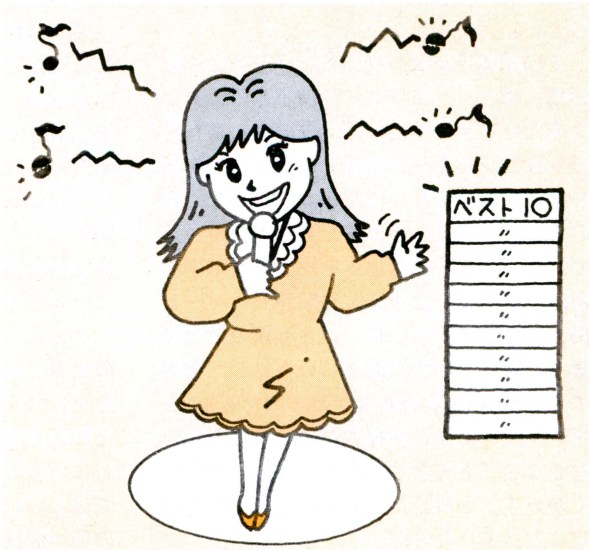
さらに娯楽における職人芸は，歌謡界にだって存在する（またこのネタかって？いいの，どうせ私はミーハーだもの）。アイドルの歌ひとつとってみても，一部ではあんなもの歌ではないと酷評されつつも，現実にはヒットチャートの上位でニッコリと笑っているのだ。

まあ，確かにアイドルタレントに音楽性を求めるのは酷である。いまはキャラクターイメージがまずは先行し，そのイメージをもとにプロジェクトチームが動き出すわけで，作曲家は音楽性よりもそのキャラクターに合ったひたすら「カッコイイ」とか「ノリのいい」コード進行やリズム，構成を追求するシステムになっている。

歌詞に関しても同じことで，まず曲が先にできていて，作詞家が「カッコイイ」とか「グッとくる」などというフレーズを入れた歌詞をくっつけるという手法がとられているのだ。しかし，たとえこういう作り方でも，ターゲットにしている人々（ミーハーなアイドルの曲であれば，それなりにミーハーな少年少女たち）を楽しませる技巧（職人芸）がしっかりしていたからこそ，ヒット曲という大きなパワーが生まれる。そしてそのパワーは，ときとして音楽性で人々を感動させるアーティストを駆逐してしまうこともあり得るのだ。

だが，やはり娯楽性にはアーティザンが必要となってくる。基本的に娯楽というのは私たちをいろいろな意味で楽しませてくれる（たとえば，ホラー映画で観客を驚かせるのと同じ）ものであり，小難しいテーマを提示して私たちを悩ませてくれるようでは困ってしまう。だから娯楽には，人を楽しませることを第一に考えるアーティザン的要素を持つことが必要とされる。

しかし，もっと物事を突き詰めて考えてみると，娯楽を作り出すタイプはアーティザンだけで十分なのだろうか，という疑問も生まれる。娯楽をもうひとつ推し進めた"娯楽文化"という枠で考えていけば，アーティザンだけが担い手だけではないような気もする。その証拠に，ゲームの世界には



もうすでに，立派にアーティストというものが存在しているのだ。

ゲームの世界のアーティストたち

ゲームの世界におけるアーティザンたちが，プレイヤーを楽しませるためにアイデアを駆使する職人だとすれば，アーティストはゲームに対して自分のポリシーを持ち，そのポリシーを実際にゲームというかたちで表現する芸術家とでもいえばいいのだろうか。芸術家なんて大げさないい方なのかもしれないが，とにかくアーティストの手によるアーティスティック(artistic：芸術的なという意味より，ここではもう少し拡大解釈して「独創的な」ということにしておこう）なゲームには，作者の個性が強く現れる。そしてゲームのアーティストは独自のポリシーを持っているから，ときとしてプレイヤー側がついて行けなくなることも多く，そういった点ではかなりアーティザンとは違ったタイプの人種だともいえる。

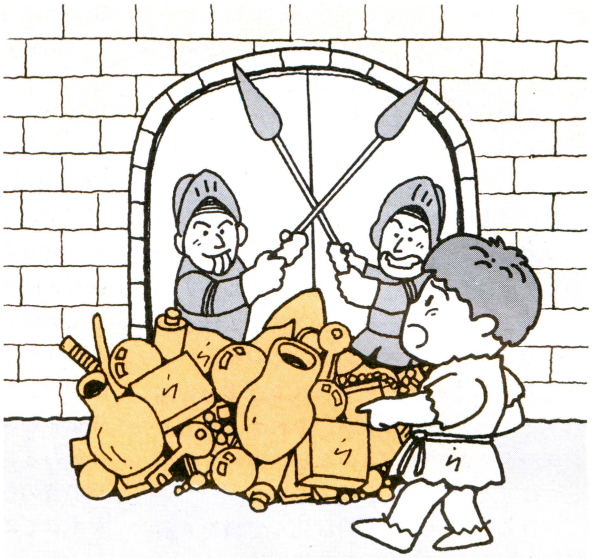
ゲーム界におけるアーティストとしての第一人者は，やはり遠藤雅伸氏だろう。あの元祖「ゼビウス」や「ドルアーガの塔」，「イシターの復活」などを制作した人物である。

1983年に登場し，爆発的なヒットとなったゼビウスは，企画の初期段階では時代設定を現代にした単純な縦スクロールシューティングゲームだった（確か「シャイアン」という仮名だったと思う）。

しかし，その後ゲームの制作に加わった遠藤氏が，アーティストとしての感性を発揮してそこに「遠藤雅伸の世界」とでもいうべき魂のようなものを吹き込んだのである。その結果，その緻密な設定と神秘性を持った独特の世界が数多くの人々（特にゲームマニアたち）を感動させ，大ヒットとなった。

その遠藤氏が次に制作した「ドルアーガの塔」は，ハッキリいって理不尽のカタマリみたいなゲームだった。先の面に進むためにはプレイヤーは山ほどのアイテムを集めなければならないのだが，とにかく，そのアイテムを見つけ出すための条件が尋常ではない。決まった地点を決まった順序で通過する。または決まった順序でレバーを倒すなどはまだ序の口で，5種類の敵を決まった順序で倒す（簡単にいうけど，全部で120通りの組み合わせがあるのだぞ），1プレイヤーボタンを押さなければならない，などなど「そんなこと誰がいったい思いつくかい！」と叫びたくなるほどのセコイ設定になっていたのだ。

しかしそれでも，このゲームは一部のマニアたちからは熱狂的な支持を受けた。おまけにファミコンに移植されたときには，業務用を上回るほどの人気ぶりであった。やはりこのゲームにも，ゼビウスとはまったく違うタイプとはいえ，独特の「遠藤雅伸の世界」がその姿をしっかりと現していたのである。



遠藤氏と並び称されるアーティストとしては，「ザナドゥ」，「ロマンシア」，「ソーサリアン」などを生み出した木屋善夫氏もそのひとりである。彼がこれまで生み出したゲームには，その端々から強力な個性がオーラのようににじみ出ているのが感じられる。

まず，「ザナドゥ」には85種類もの武器やアイテムが用意されていたが，それらすべてに経験値が設定されていた。よくよく考えてみれば，よくもまあこんなムチャクチャなことが思い浮かぶものである。うっかりデータ量を増やそうものなら，プレイヤーの負担が大きくなるだけで，ユーザーフレンドリなんぞという言葉とは無縁のものでしかない。

ところが現実とは恐ろしいもの。実際には，すべてのアイテムの経験値を最大にしようと奮起する好き者が数多く現れたりして，結局は大ヒットとなったのである。

これらはすべてイジワルさ（本人の性格の悪さだと酷評した雑誌もあったが）以上に，そこに形成された「木屋善夫の世界」が，多くのファンを魅了してしまった結果だといえる。

さらに彼はその後「ロマンシア」で追い打ちをかけてくるのだが，これまた「イジワル木屋」=「独創的世界」というイメージを定着させてしまうほどの強烈な個性をこのゲームによって放ち，完全に彼なりの世界を完成させてしまった。そしてその最新版ともいえるのが，現在，若い世代の熱狂的支持を受けている「ソーサリアン」のシステムである。

このソーサリアンは，木屋氏の持つ発想の世界を前提にまず構築され，そしてそれを土台にしてシナリオやモンスターといったRPG的要素を付随させて完成させ，ゲームにいまのようなリアリティを持たせたのではないかと私は思っている。それだけ氏の持つ個性は，特徴的かつ独創的なものではないのだろうか。

今度は海外に目を向けてみよう。やはりその代表格といえば，ロバート・ウッドヘッドと並び称される「ウルティマシリーズ」のロード・ブリティッシュもそのひとりだろう。国民性や宗教的背景があるとはいえ，「剣と魔法」だけに終始してしまうRPGの世界において，「徳を積んでアバターになる」といったことをプレイヤーの最終目的に選んだのは彼独自の発想であり，また彼の持つ世界観の主張ともいえるのだ。まっ，このゲームが日本人に対してどうだったか，などという意見はあえてここではしないでおこう。

明日はどっちだ

さて，このようにゲームの世界で活躍する既存のアーティストたち。しかし，最近の日本のゲーム界においては，イースの例を見るまでもなくアーティストよりもアーティザンの手によるゲームのほうが受け入れられつつあるのだと私は思っている。しかし，とどまることを知らないゲーム界において，果たしてその次にくるものはいったいなにか？

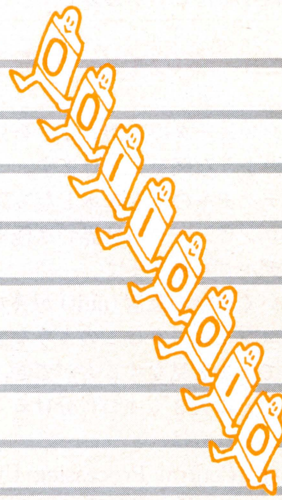
今回はたいへん長い序章となってしまった感もあるが，それはご勘弁していただくとして，次回はこの仮説に対していよいよ事態は核心へと向かう。今回の仮説が果たして実証できるのか。その根拠はいったいどこにあるのか。はたまた，これからのゲーム界に君臨するアーティザナルの実態とはいったいいかようなものであるのか。

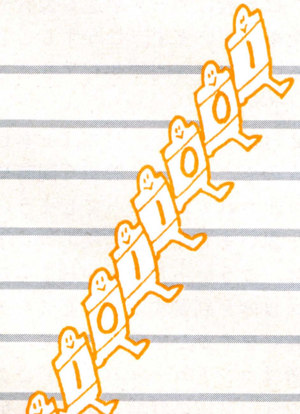
予断や不平不満を一切許さないような大風呂敷を広げつつ，私なりの解答（決着）目指して，皆さんを道づれに突入して行くのです。

特集

マシン語“でじたるざんまい”

“マシン語”，このなんともいえない強力な響きに多くのパソコンユーザーが心を奪われ，立ち向かい，そして挫折していきました。マシン語を意のままに扱えるようになることは，ハードウェアを自由にドライブできることを意味しています。だからこそマシン語は，ユーザーにとって，ドラゴンを倒すことにも匹敵するのです。そしてドラゴンが成長するように，マシンもまたグレードアップしてユーザーの前に立ち塞がります。かつては素手で戦ってきた勇者も，その手によい武器を持たなければなりません。そう，マシン語入門にはよいアセンブラが必携のアイテムとなります。今月は，Z80用に新しい高速エディタアセンブラを発表することができ，ここにマシン語入門の一大特集となりました。また，マシン語はCPUに依存する言語であり，Oh! XではZ80と68000の2つのCPUについて扱うことになります。ただ，マシン語入門のために理解しなければならないコンピュータの基本的なしくみはそれほど変わるものではありません。今回の特集では，CPUとマシン語の関係に始まり，アセンブラ一般の基礎知識，そしてZ80マシンとX68000のそれぞれに実践的入門のコーナーを設けました。Oh! Xは新たな勇者の登場を期待しています。





アーキテクチャからのマシン語入門

Sohma Hidetomo 相馬 英智

マシン語は基本的に，CPUに対する命令です。そこで今回の特集では，実際のプログラミングに入る前に，CPUを中心とするコンピュータのアーキテクチャを理解することから，マシン語入門のアプローチをかけてみたいと思います。初心者の皆さんにも，いろいろと役に立つと思いますのでぜひ読んでみてください。

コンピュータのアーキテクチャ

コンピュータにおけるアーキテクチャとは，コンピュータのハードウェア的な構成のことをいいます。ただ，それは論理回路がどうなっているかという回路図みたいな細かいものではなく，もっと大雑把にとらえます。したがって，ハードウェアそのものについての知識は特に必要ありません。むしろ柔軟な感性こそが必要です。

コンピュータのイメージモデル

よくコンピュータの入門書などを見ていると図1のような図を見かけます。この図はアーキテクチャの最も基本となる図のひとつで，コンピュータの世界では極めて伝統的なモデルでもあります。

図にはいくつかの装置(unit)が書かれています。コンピュータはこのような装置で構成されていると見なすことができ，それらは目的ごとに大きく4つに分けることができます。それらは，

中央処理装置 (Central Processing Unit：略して“CPU”といいます)
記憶装置 (Memory)
入力装置 (Input Unit)
出力装置 (Output Unit)

の4つです。

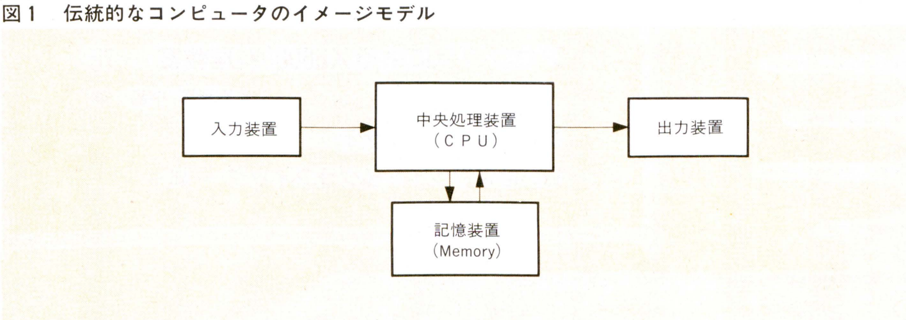
キーボードやディスプレイなどの入力装置や出力装置はコンピュータの外側にあり，目につく位置にあるのでわかりやすいと思います。それに対し，記憶装置は半導体メモリやディスクドライブなど，必ずしもコンピュータの外側にはありません。CPUに至っては，ほとんどの場合外側からは目につかないようになっています。

記憶装置はデータを格納しておくための装置，CPUは演算などのデータの加工を行うためのもので，コンピュータの心臓部に当たります。それゆえCPUの能力(パワー)はコンピュータの能力を決定する重要な要素となります。

さて図1には，これら4つの装置が矢印で結ばれています。もう皆さんにはおわかりでしょうが，この矢印はデータの動きを示しています。つまり入力装置でユーザーからデータを受け取り，CPUはそれを記憶装置に出し入れしながら，最終的な結果であるデータを出力装置に出力するというのが，コンピュータなのです。

ただし，図1のモデルにはコンピュータの構成についての重要な情報が欠けています。それはCPUはデータの処理を行うだけでなく，入力装置や出力装置などコンピュータを構成する大部分の装置の制御をも行うということです。つまり，CPUはデータの処理（具体的には演算）と各装置の制御という2つの役割を担っているわけです。

図1 伝統的なコンピュータのイメージモデル



CPUはコンピュータの中枢であることがおわかりでしょう。

バスの概念

さて，図1のモデルはわかりやすさに重点を置きすぎているので，実際のコンピュータ内部の構成とはかなり異なっています。そこで，より本物に近いコンピュータのモデルが図2で，各装置を結ぶのは**共有バス**と呼ばれるものです。

バスは，具体的には複数の電線の束です。各装置はこのバスを使って，データのやり取りを行います。そして各装置はこのバスにぶら下がっています（バスに“つながっている”というよりは“ぶら下がっている”というのが通みたいです）。

一般的に共有バスは，目的によって，

データバス
アドレスバス
コントロールバス

の3つに大きく分けられます。

このうち，データバスとアドレスバスはマシン語入門のためにはぜひ理解しておきたいものです。まず，データバスは文字どおりデータを転送するために使うもの。アドレスバスは一般にデータを送る装置や場所を示すために使います。こうすることでバスをいろいろな装置間で共有することができるわけです。

さて図2のような場合，バスは1本しかありませんので，2つの装置から同時にデータが送り出されてぶつかってしまうのではないかと思いませんか。最後のコントロールバスは，こういったことが起こらないようにバス自体を制御するのに必要なデータの受け渡しをするためのものです。こうして各装置はコントロールバスなどによって，バスや他の装置の状態を知ることができ，確実な装置間のデータの転送が確保されるのです。

一般的なCPUのアーキテクチャ

CPUはコンピュータの中枢であり，最も重要な働きをします。そこで，このCPUのアーキテクチャについて見ていきたいと思います。

CPUの構造

一般にCPUは，制御とデータの処理（演算）の２つの作業を行います。まず図３を見てください。これがCPUの最も基本的な構成です。この図によるとCPUは制御を行う部分（ここでは**制御装置**と呼ぶことにします）と，**ALU**（Arithmetic Logical Unitの略：演算装置）と，**レジスタ**と呼ばれる記憶装置，そして**バス**で構成されていることがわかります（ただし実際のCPUはこれら以外にも多くのもので構成されています）。

それぞれの役割から見ていきましょう。

●制御装置

制御装置は，CPU内部の制御を行うところです。具体的には，ALUの動作，CPU内のバスの制御などです。さらに，この制御の結果はCPU内部はもちろんのこと，先ほどの共有バスのコントロールバスにまで反映されます。したがって，CPU以外の装置は，このCPUの制御装置から出るコントロールバスを調べることで，CPUの状態などを知ることができます。

●ALU

次にALUですが，これは算術的な演算や論理的な演算などを行います。図３のALUを見ると，入力が２つで出力が１つとなっています。したがってALUは１つまたは２つのデータをバスから受け取り，演算して結果をバスに出力します。

●レジスタ

そしてレジスタですが，これは小規模な記憶装置です。この小規模とは記憶できるデータの最大量が，比較的少ないことを意味します。その代わりこのレジスタは，コンピュータの記憶装置のなかでも最も高速にデータの入出力が可能です。

このレジスタは，CPU内で処理を施すデータや処理の結果，そしてCPUの状態などを記憶（格納）しておくためのものです。通常複数のものがあり，使う目的が定まっているレジスタと，いろいろな使い方をすることが許される**汎用レジスタ**と呼ばれるものがあります。

また，ALUでの計算の際に頻繁に用いられ，計算の結果などが格納されるレジスタを特にアキュムレータ（Accumulator略してAcc）といいます。レジスタにはそれぞれに名前（簡単なアルファベット）が付いていて，それで区別されています。

●内部バス

そして，最後にバスです。これは先ほどまで話していた共有バスとは，性格が異なります。区別のため，今までのCPUやI/O装置などをつないでいたバスを外部バス，CPU内のバスを内部バスと呼んでいます。この内部バスによって，制御装置，ALU，そしてレジスタはデータの転送などを行います。

マシン語による制御

これらの役割がわかると，CPUの動きが見えてきます。たとえば制御装置が，あるレジスタ（これをレジスタＡとします）の値にレジスタ（レジスタＢ）の値を加えて，さらにもとのレジスタ（レジスタＡ）に格納したいとします。すると制御装置は以下のように動きます。

1)　２つのレジスタＡとＢから値を内部バスに押し出す。

2)　ALUが内部バスからデータを受け取り，足し算を行い，その結果を内部バスに出力する。

3)　内部バスにのっている値をもとのレジスタＡに格納する。

これで目的の計算ができるわけです。さて，制御装置にこのような指示をするのは誰（何）でしょうか。それが私たちが書くマシン語のプログラムなのです。

マシン語を使うと，先ほどの３つの過程が１つの命令となります。具体的には（厳密にはCPUによって書き方が異なる），

ADD　A，B

といったふうになります。

記憶装置

では，CPU（の制御装置）を動かすプログラムはどこにあるのでしょうか。また処理するデータはどこにあるのでしょうか。

これらは，すべて記憶装置（Memory：メモリ）に格納されています。記憶装置といってもさまざまなものがあります。テープやディスクドライブ，そして半導体メモリなどがそうです。

記憶装置は大きく２つに分けることができます。それは，CPUが直接データのやり取りを行う主記憶と呼ばれるものと，間接的にデータのやり取りを行う補助記憶装置と呼ばれるものです。

一般に主記憶装置には，高速にデータのやり取り（これをデータのアクセスといいます）ができる半導体メモリなどを（それで主記憶のことをメモリと呼んだりします），補助記憶装置には比較的低速なアクセスしかできないテープや磁気ディスク，光ディスクなどを用います。ちなみにCPU内のレジスタ（CPU以外の装置も持っている

図２　共有バス

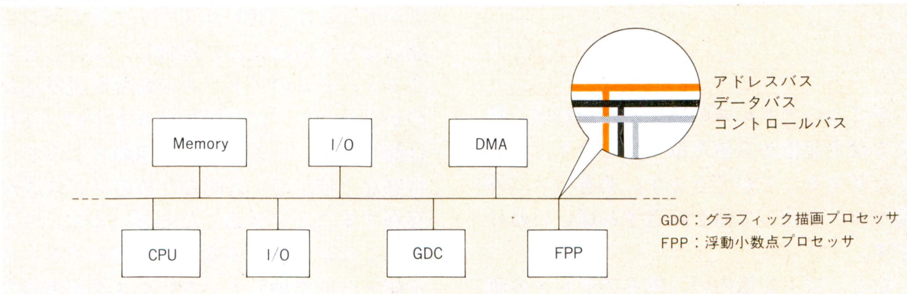


図３　CPUの構成

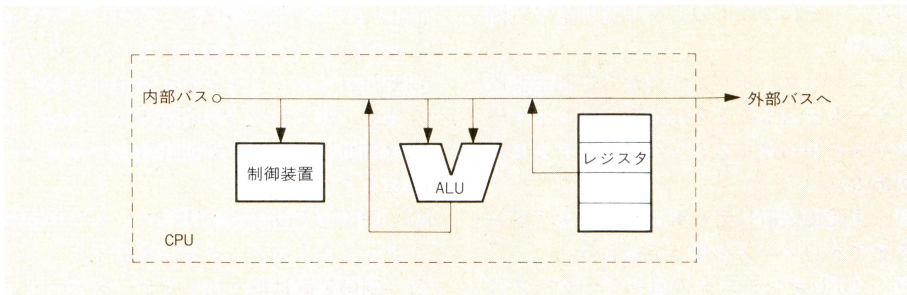
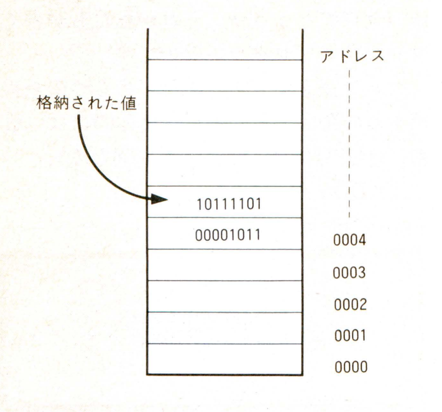


図4



データか？ 命令か？

さて，プログラムとデータを同じ主記憶に置いてしまうやり方を“ストアードプログラム方式”とか“ノイマン方式（ノイマンは人の名前：von Neumann)”などといいます。では主記憶にあるプログラムとデータは区別できるかというと，マシン語レベルでは本質的には不可能です。また，データといっても，数値はもちろん文字などさまざまなものがあるわけですが，これらの区別さえ行われません。というのは，コンピュータの内部では，すべての情報が1と0の並び（ビット列）といった形でしか表現されないからです。

たとえば主記憶のあるところに“10110011”という値があったとすると，これをマシン語の命令と解釈すればある命令となり，数値と解釈すればある値となり，文字と解釈すればある文字となるといった具合です。したがって主記憶にあるビット列の値は，私たちが予期した（望んだ）ように解釈されないと，大変なことが起きてしまいます。しかし逆に，プログラムが処理をしている途中で，自分自身をデータと見なして書き直すことにより，自分自身で変化していくプログラム（うまくいけば成長するプログラム）を書くことができることになります。このノイマン方式は長所と短所を同時に供給しています（人生なんてそんなものだよ）。

場合がある）は，極めて高速にアクセスできる半導体メモリの一種です。

アドレス

さて主記憶には，これらのプログラムやデータがどのように格納されているのでしょうか。先ほど，レジスタがそれぞれ名前を持っていることを話したと思います。同様に，主記憶もプログラムやデータが入っている場所を区別できればいいわけです。そこで出てくるのが**アドレス**(Address)です。

アドレスとは番地のことで，つまりデータなどが入っている場所に，○○番地という番号を振ってしまおうというわけです。どういうふうに番号を振るかというと，図4のように単純に0から順番に振っていきます。このとき一般的にアドレスの番号が大きいほうをアドレスの上位，少ないほうを下位と呼んでいます。

データの入出力

さてこのアドレスの概念がわかれば，私たちは主記憶から値を取り出したり，格納したりすることができます。共有バス（外部バス）を使って値を転送すればいいわけです。

たとえば主記憶内の，あるアドレスの値をCPU内のレジスタに格納するとします。すると，それは以下のようになります（図2参照)。

1) コントロールバスで今から主記憶装置にデータを送ることを伝える（コントロールバスの中にはこのような働きをするものがある)。

2) 主記憶装置にデータを転送するアドレスをアドレスバスを使って伝える。

3) CPUがレジスタの値をデータバスに送り出す。

4) 主記憶装置がデータバスのデータを，示されたアドレスに格納する。

これらも，先ほどの例と同様に1つのマシン語の命令で行われますので，バスがどうとかということは私たちは気にする必要はありません。

さて，これで主記憶（メモリ）のイメージがわかってもらえたと思います。今までは，CPUとメモリの間のデータの転送でしたが，他の装置とCPUの門，あるいは他の装置とメモリについてもまったく同様に行われます。

CPUの動作とレジスタの機能

ここまでくれば，私たちはCPUがどのように動作しているかを，より正確に知ることができます。CPUはメモリにあるマシン語の命令を読み込んで解釈し，実行します。すると，どこのアドレスの命令を読み込むのかということが問題になりますが，これは簡単です。レジスタの説明のところで，特殊な（特定の）目的のためのレジスタが存在することをお話したと思います。つまり，これから実行する命令のあるアドレスを値として格納しているレジスタがあるのです。これが一般に，**PC**(Program Counter)**レジスタ**と呼ばれるレジスタです。なおこのPCレジスタは，ほかにもいろいろな名前が付けられていますので注意が必要です。

さて，PCレジスタの存在がわかれば，あとは簡単です。CPUの動作は以下のように書けるでしょう。

1) CPUの制御装置がPCレジスタの値をアドレスとするデータを取り出す。

2) 制御装置は取り出したデータをマシン語の命令として解釈する。

3) PCレジスタの値を1命令分だけ増やす（次の命令のあるアドレスになる)。

4) 解釈したマシン語の命令を実行する。

5) 1)へ行く。

このように，CPUは単純なループを回っているわけです。一般に上の1)，2)のようにマシン語の命令の取り出しや解釈を行う段階を，**命令取り出し段階**（Instruction fetch cycle)，4）のように命令の実行を行う段階を，**命令実行段階**（instruction execution cycle）と呼んでいます。

また上のことから，マシン語のプログラムが主記憶の一部に連なって続いているように見えます。確かに普通はそうなのですが，BASICのGOTOやGOSUBに当たるようなマシン語の命令がありますので，ぜんぜん違うアドレスにあるプログラムに移ることがあります（そのような命令はPCレジスタの値を書き換えることで実現されます)。

さまざまなレジスタ

CPUのレジスタには，**汎用レジスタ**と呼ばれるいろいろな使い方ができるものと，先ほどのPCレジスタのように特定の目的のためだけに存在するものの2つに分けられると思います。

まず汎用レジスタは，コンピュータ内のいろいろなデータが格納できるように作られたもので，何組か存在します。一般に，アキュムレータなどのレジスタは，最も頻繁に使われるレジスタなので，広い意味では汎用レジスタとなっています。

これに対して，特定の目的用のレジスタにはさまざまなものがありますので，そのうち特に重要なものについて説明しておきたいと思います。

インデックスレジスタ

先ほどのPCレジスタは，実行すべきマシン語のプログラムのあるアドレスを格納していましたが，このようにアドレスを主に格納するためのレジスタがあります。具体的にはインデックス（Index）レジスタなどがそうです。

インデックスレジスタは，使いたいデータのあるメモリのアドレスを格納しておいて，そのアドレス付近を簡単に扱えるようにするものです。つまり，インデックスレジスタに格納したアドレスから3つ前とか5つあととかいうふうにして，目的のアドレスを指定するわけです。

このように使用したいメモリのアドレスを示すには，直接そのアドレスをマシン語の命令のなかに書いておく方法，レジスタ（この場合はレジスタは変数のようになります）にアドレスの全部または部分を格納しておいて使う方法，そして，具体的に指定したメモリに格納された値が目的のアドレスになる方法などがあります。

そこで，最初の直接アドレスを命令などのなかから指定する方法を**直接アドレス指定**，次のレジスタなどで指定するアドレスを部分的に表現する方法を**レジスタ修飾**（修正），最後のメモリなどで使う方法を**間接アドレス指定**といっています（これらについては図5や右下の囲み記事を参考にしてください）。

また，このインデックスレジスタに似たようなレジスタ（原理は似ているが，使い方はけっこう異なる）に，セグメントレジスタや，ベースレジスタと呼ばれるものがあります。これらの詳しい動作は各CPUについて，それぞれ異なりますので，そのつど勉強してほしいと思います。

図5　直接アドレス指定とレジスタ修飾

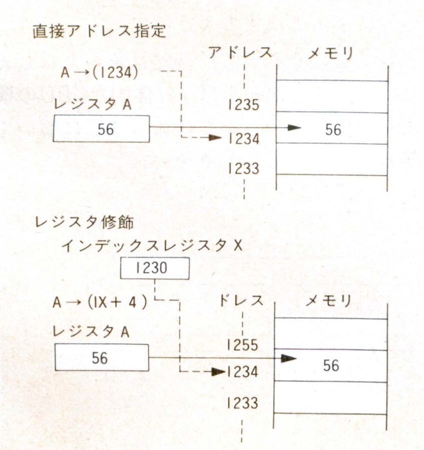
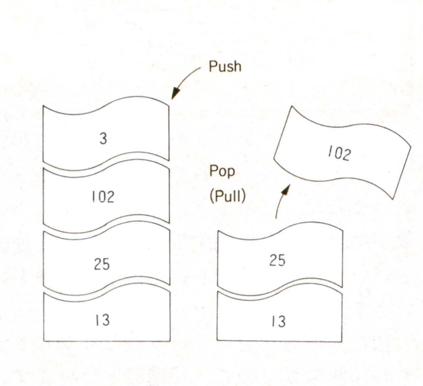


図6　スタックの仕組



スタックポインタ

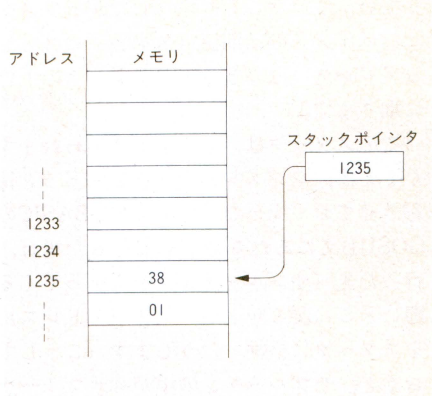
アドレスを格納するレジスタには，もうひとつ面白いものがあります。これは**スタックポインタ**（Stack Pointer）というものです。

スタックとは“積む”という意味で，コンピュータの世界ではスタックとは，データを積み重ねて管理するというデータ構造のことをいいます。といっても，なんのことかさっぱりわからないと思いますので，具体的にはどのようなものかを説明しましょう。

まずは図6を見てください。どうです，文字どおりデータが積み重なっています。これをスタックといいます。こういうふうにデータを積み重ねた状態で，保管・管理をするのがスタックのやり方です。

このスタックには，**プッシュ**（Push）と**ポップ**（Pop）の2つの操作を行えます（ポップをプル（Pull）ともいいます）。図6のように，プッシュは新しいデータをさらに積み重ねるという操作で，ポップは積み重ねられたデータの一番上のものを取り出す操作です。

図7　スタックポインタ



このようにスタックの一番上は特別な扱いを受けますので，これを特に**スタックトップ**（Stack Top）と呼びます。どうです，簡単でしょう。大昔のCPUを除くと，大体のCPUがこのスタックを持っています。そしてCPUがスタックを実現するために，欠かすことができないのがスタックポインタなのです。

図7を見てください。どのようにスタックが実現されるかを説明しましょう。準備は，スタックポインタにスタックの底となる適当なアドレスを格納するだけです。そこで，アドレスが出てきたということはメモリを使うということで，スタック内のデータはメモリに格納されることとなります。これがわかれば，あとは簡単です。

データをプッシュする場合は，“スタックポインタの値（アドレスが格納されている）をひとつ減らして”，スタックポインタで示されるアドレスにデータを格納します。逆にポップする場合は，スタックポインタで示されるアドレスのデータを取り出し，“スタックポインタの値をひとつ増やす”わけです。こうすることでスタックが実現されるわけです。

ここで注意してほしいことは，スタック

アドレス指定のテクニック

直接アドレス指定とレジスタ修飾で，特に重要な違いとなるのは，具体的なアドレスが決定される時間の差です。

つまり直接指定ではアドレスの値が命令の中などに含まれていますので，最初（プログラムができたとき）からデータを格納すべきアドレスは決まっていて，マシン語の命令の中に明示されています。これに対しレジスタ修飾は，具体的なアドレスが決まるのは命令が実行されるときで，命令の中に示されているインデックスレジスタなどの名前と，やはり命令の中に示されている“その3つ前”などといった差という2つをもとに，この命令の実行時にアドレスが計算されます（図5参照）。したがって命令実行時のインデックスレジスタの値（アドレスが値として格納されています）によって，実際にデータをアクセスするアドレスが異なっているわけです。

このことは，かなり面白い結果を示します。たとえば，メモリ上に取り込みたいデータが並んでいるような場合を考えてみましょう。

この場合，インデックスレジスタの値を1ずつ増やしながら±0のレジスタ修飾でデータをレジスタに取り込むようなループを書けば，いちいちアドレスをいう必要なしにデータの取り込みが行えます。

はアドレスの上位から下位に向かって成長していくということです。したがってスタックポインタの初期値（スタックの底となるアドレス）はできる限りアドレスの上位に取ります。

このスタックは，とてもよく使われます。たとえば，レジスタのデータをちょっと待避させておく場合などです。またBASICのGOSUB文にあたる命令を実行した場合，行った先（サブルーチン）から戻ってくる際にどこに戻るかという情報（アドレス）をスタックに格納したりします（こうしておくと，サブルーチンの中からサブルーチンを呼んでも，戻るべきアドレスが混乱しません）。

フラグレジスタ

このようにアドレスを格納するレジスタには有用なものがあるのですが，アドレス以外の特定の情報を格納するレジスタもあります。この特定のデータを格納するレジスタの中で有名なのが，**フラグレジスタ**（コンディションコードレジスタ：CCR，ステータスレジスタ：SRともいう）です。

フラグレジスタは，プログラムで行った演算などの結果の一部や，CPUの状態などが示されているレジスタです。このレジスタはマシン語の命令が実行されることで，内容がどんどん更新されていきます。そして私たちはこのフラグレジスタを見ることで，演算がうまくいったかどうか，CPUの状態はどうなっているかということを知ることができます。

フラグレジスタの細かい動作はCPUによって異なりますので，それらについては自分のマシンのCPUについて勉強してください。はっきりいって，このフラグが理解できていないとマシン語は使えないといっても過言ではありませんので，しっかりと勉強してください。

図8

2進数	16進数	10進数
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

コンピュータが扱うデータ

前の節で，コンピュータ内のすべての情報は1と0で表現されているということをいいました。このように，1と0の2つの数字だけで値を表現するやり方を，2進法といい2進法で表現された値を2進数といいます。

私たちが通常使っている数は0から9までの10個の数なので，10進数といいます。0から1ずつ加えていったとすると，10進数は9の次は1桁上がって10になります。同様に2進数では，1の次は1桁上がって10になります。2進数の数と10進数の数の対応を図8に示します。これを見ると2進数がどのように増えていくのかということがわかると思います。

2進数の扱い方

まず，2進数と10進数の対応を簡単に計算するためには，次のようにします。基本的に10進数は以下のような特性を持っています。

$$538=5\times10^2+3\times10^1+8\times10^0$$

当たり前と思われるかもしれませんが，これはとても重要なことです。これを使えば，2進数を簡単に10進数に変換することが可能です。たとえば，

$$\begin{aligned}1011&=1\times2^3+0\times2^2+1\times2^1+1\times2^0\\&=8+0+2+1\\&=11\end{aligned}$$

となります。10の代わりに2を使うのがミソです。

さて，2進数というのは，私たちにとってあまりにも使いづらいので，通常は16進数の力を借ります。どうして16進数を使うのかというと，これが2進数と相性がいいからです。図8を見てもらえばわかるように，4桁の2進数と1桁の16進数がピタリと対応しています。

16進数では16個の数字が必要ですから，0から9までの数字にAからFのアルファベットを加えて使います。なお，2進数，10進数，16進数が混在すると区別がつかないので，2進数を表す場合には数字のあとに“B”を，16進数を表すときは“H”をつけてたりします。

ビット，バイト，ワード

2進数などの話は数学っぽい話でしたが，これからはもっとコンピュータっぽい話をしましょう。

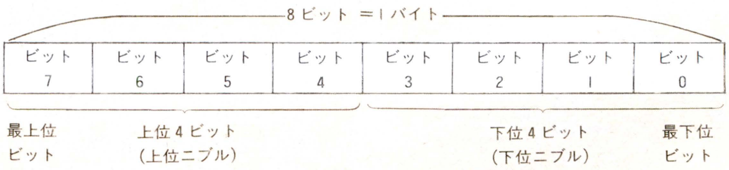
まずは用語の定義からです。1桁の2進数の数字の1もしくは0というデータは，コンピュータで処理されるデータの最小単位であり，これを**1ビット**（bit）と呼んでいます。したがって4桁の2進数“1001”というのは4ビットになります。

一般にコンピュータでは，8ビットをまとめて処理することが多く，これを基本単位として**1バイト**（byte）といいます。図9を見てください。このように1バイトのデータには，それぞれのビットごとに名前が付けられています。つまり一番下の桁の1ビットをビット0，それから順番にビット1，ビット2と続き，一番上の桁の1ビットがビット7です。最初がビット0というふうに0から始まることに注意してください。また特に，ビット0を最下位ビット，ビット7を最上位ビットと呼んでいます。

この8ビットは16進数で表現すると2桁になるのですが，その16進数の下の桁にあたる4ビットを下位4ビット，上の桁にあたる4ビットを上位4ビットと呼んでいます。また，これらのビット，バイトのほかに**ニブル**（nibble）と**ワード**（word）というものがあります。まず，ニブルは4ビット（半バイト）のことです。これは16進数で表現すると1桁分となります。またワードはふつう16ビットか32ビットのことを指します。どちらを指すかはケースによります。これはコンピュータ（CPU）のデータの使い方によって決まるようです。

さてこのぐらいで，基礎は終わりにしましょう。次の節からは，具体的なCPUの構成や基本的なマシン語の命令などについて触れてみることにします。

図9



Z80とマシン語

Z80は８ビットCPUの代表格であるといえるザイログ社のCPUであり，インテルの8080に，新しい命令やレジスタを加えるなどして拡張したものです。

Z80のレジスタ構成

まずは図10を見てください。これはZ80のレジスタの構成です。

Z80は，**表レジスタ**と**裏レジスタ**というまったく同じ２つの汎用レジスタセットと，特定の働きを行う１つのレジスタセットを持っています。ただ，汎用レジスタセットは一度に表と裏のどちらかしか使えません。これはカードの表と裏と同じで，一度に両面は見れないわけです。そこで現在使われているほうを表レジスタ，使われていないほうを裏レジスタと呼び，裏レジスタのレジスタには「′」(ダッシュ)を付けて示します。

実は，この表裏をひっくり返す命令があります。ここで，レジスタＡとＦだけは，ほかのＢからＬまでのレジスタとは独立にひっくり返すことができるのです。これは，あとで述べる命令よりは使用頻度が低いのですが，縁起物なのでいいますと，

EX　AF，AF′命令

でＡ・Ｆのレジスタの表裏をひっくり返します。また，

EXX命令

で，ＢからＬまでのレジスタの表裏がひっくり返せます。

これに対しＩ，Ｒなどのレジスタセットはいつも使うことができますが，これらは特殊な目的のために使いますので，汎用レジスタセットとはかなり感じが違います。そこで今度は，それぞれのレジスタの働きについて見てみましょう。

●アキュムレータ

Ａ，A’レジスタは，いわゆるアキュムレータです。つまり算術演算や論理演算などの処理は，このレジスタのデータについて行われ，その結果もこのレジスタに格納されます。このレジスタが，Z80では最も使われるレジスタで中心的な役割を示します。

●フラグレジスタ

Ｆ，F’レジスタはフラグレジスタです。したがって，CPUが行った演算などの実行結果の一部がここに表現されます。

このＦレジスタは８ビットなのですが，このうち６ビットが１ビットずつそれぞれ意味を持っていて，それをフラグといいます。図11に，フラグレジスタＦの構成を示します。ここでは，このうち特に重要と思われるＺ，Ｃ，Ｓの３つについて説明します。

まずＺフラグは**ゼロフラグ**（Zero Flag）と呼ばれ，演算した結果が０になったときだけ１となり，そうでないときは０です。なおこのように，フラグが１となることをフラグが立つといいます。

次にＣフラグは**キャリフラグ**と呼ばれるものです。ここでいうキャリ(Carry)とは足し算の繰り上がりのことです。たとえば，８ビットデータの足し算をしたとき，結果の値が大きくなってしまいアキュムレータ(この場合はＡレジスタ)に入らなくなった場合に立ちます。また，引き算の際の繰り下がり（ボロー：Borrow）のときも立ちますし，Z80ではほかにローテイト命令，シフト命令などの場合にも変化し，またSCF命令というキャリフラグに対する特殊な処理の場合にも変化します。

最後にＳフラグですが，これは**サインフラグ**と呼ばれ，演算結果が（アキュムレータの値が）負の数のときに，このフラグが立ちます。さて負の数とは，２の補数表示と呼ばれる表現での負の数のことで，具体的には最上位ビットが立っていると負の数となります。なおマシン語では正の数であることをＰ（Plusの略），Ｍ（Minusの略）と書きます。

●汎用レジスタ

Ｂ～ＬとB’～L’の６個２組の汎用レジスタは，汎用レジスタの名前どおりいろいろな使い方ができるレジスタです。そこでこれらの使い方は大体，ユーザー任せという感じになります。

ちょっと注意がいるのは，Z80には**レジスタペア**という考え方があります。これは８ビットのレジスタ２つを組み合わせて，16ビットのレジスタとして使おうといったものです。ただ，８ビットのレジスタならなんでも組み合わせられるかというと，そうではありません。図10を見ると，８ビットのレジスタが横に２つずつ並べて書かれています。これがレジスタペアの組み合わせです。つまり，AF，BC，DE，HLというレジスタペアがあるわけです。

レジスタペアの名前はレジスタの名前を組み合わせただけの簡単なものになっています。これらのうちBC，DE，HLのレジスタペアは汎用であり，そのうち特にHLレジスタペアは重要です。というのは，このHLレジスタペアは加算と減算という算術演算のできるアキュムレータとしての機能を持っているからです。ただし，加算や減算のときの相手のレジスタはレジスタペアなどの16ビットのレジスタに限られます。これについて詳しくは，このあとのマシン語の基本的な命令の説明のところで触れたいと思います。

●その他のレジスタ

そして残るレジスタセットなのですが，ここのレジスタについてはすでに説明したものがたくさんあります。PCレジスタ(プログラムカウンタ)，SPレジスタ(スタックポインタ）については「CPUの動作とレジスタの機能」のところで述べたとおりです。またIX(Index X)レジスタ，IY(Index Y)レジスタはインデックスレジスタで，これについても先ほど述べたとおりです。

残るＩレジスタは**インタラプトレジスタ**と呼ばれるもので，I/O装置などがデータを送りたいときや，特殊な処理を行いたいときに“割り込み”という技法を使うのですが，この場合に使うものです。かなりマシン語が使えるようにならないと，Ｉレジスタを使うことはないと思われるので，初

図11

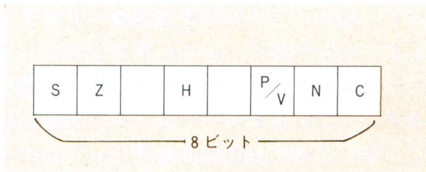


図10　Z80のレジスタセット

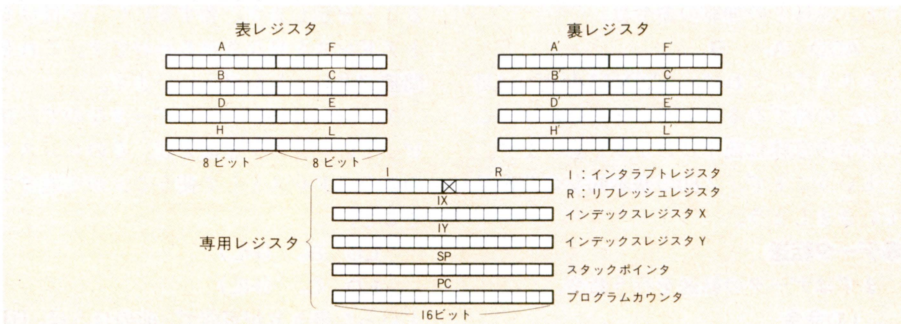
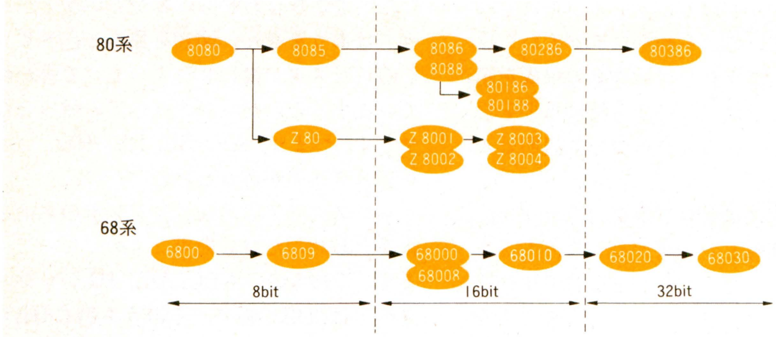


図12　CPUの系譜



めのうちは気にしなくてもよいでしょう。

またRレジスタは，**リフレッシュレジスタ**と呼ばれるものです。これは，メモリとして半導体メモリのダイナミックRAMを使う際にCPUが使うレジスタで，私たちが直接使うことはめったにありません（ただ，乱数の種を取るときには便利だったりします)。このレジスタは7ビットの幅しか持っていません。ということでレジスタについての説明は終わりです。

Z80の命令

では，Z80の簡単なマシン語の命令について，説明したいと思います。CPUが本来持っているマシン語の命令を，命令セット（Instruction Set）といいます。先ほど説明したと思いますが，マシン語のプログラムも1と0のビット列ですから，これを直接に理解するのは極めて難しく，よほどうまくやらないと混乱するだけです。

たとえば，アキュムレータAの値にレジスタBの値を加える命令が“10000000B”であるといっても，イメージが湧かないと思います。また全命令セットを暗記するのはあまりにもバカバカしすぎます。そこでマシン語の1つの命令を意味をとりやすくして，わかりやすいようにするものとして，ニーモニック表記というものを普通は使います。ここでもそのニーモニック表記を使い，特にZ80では標準的なザイログ社のニーモニックを使います。たとえばこれを使うと，先ほどの加算の命令は，

ADD　A,　B

となります。これは，ADDがAddition（足し算）の略であることを知っていれば，この命令の意味は簡単にわかると思います。

ということで，Z80の基本的な命令を見ていきましょう。

●データ転送

まずはデータの転送を行う命令，

LD命令

です。これはLoaDの略のLDと書き，さらにLDのあとに2つの要素を書きます。するとLDのあとの最初の要素へ次の要素からデータを転送しなさいという命令になります。たとえば，

LD　A,　B

で，レジスタBのデータをレジスタAに転送することを示します。なお，データはコピーされるわけで，レジスタBのデータはそのままです(変数と同じ)。また，これはレジスタからレジスタへの転送なのですが，この場合，レジスタペアについてはできません。たとえば，

LD　HL,　BC

というのはありません。しかし以下のように書けば，同じ効果が得られます。

LD　H,　B
LD　L,　C

さて，レジスタに直接値を格納したい場合には，

LD　A,　10H

というふうに書きます。これはレジスタAに16進数の10（10進数の16）を値として格納することを意味します。こういうふうに直接データ（値）を書く方法を**イミディエイト**（即値）**アドレッシング**といいます。

次に，メモリとデータのやり取りをしたい場合は，

LD　A,　(1234H)
LD　(3568H),　A

というふうに書きます。前のほうは，レジスタAの値を，アドレスの1234Hに転送することを示します。要するに( )を付けるとアドレスと見なされるわけです。これを**間接アドレッシング**といいます。

アドレスは16ビットのデータなので，16ビットのレジスタを使えば，そのレジスタの値をアドレスとして使うことが可能です。そこで，

LD　B,　(HL)
LD　C,　(HL)

といった書き方が可能で，前のほうは，HLレジスタペアに格納されている値をアドレスとして，そこに格納されたデータがレジスタBに転送されるわけです。インデックスレジスタを使えば，

LD　D,　(IX+3)
LD　(IY+5H),　A

と書くこともできます。これはわかりますよね。

これで転送の命令は終わりですが，アドレッシングの使い方（かっこの意味など）は，これ以外の命令でも使えますので注意してください。

●算術演算/論理演算命令

次に，算術演算命令です。これには，基本的なものとしては，

ADD命令　（加算を行う）
ADC命令　（キャリフラグを含めて加算を行う）
SUB命令　（減算を行う）
SBC命令　（キャリフラグを含めて減算を行う）
INC命令　（1だけ増やす）
DEC命令　（1だけ減らす）

といったようなものがあります。

これらはたとえば，

ADD　HL,　BC
SUB　(HL)
INC　B
DEC　(IX+9)

というふうに使います。

これらの命令は単純に計算するだけでなく，計算の結果の一部をフラグレジスタFに反映させます。つまり計算結果がちょうど0ならば，Zフラグが立つといった調子です。

また，フラグを動かすことによってデータを調べるといった命令もあります。それが，

CP命令　（データの比較を行う）

で，これはレジスタの値を変えることなく引き算をしたような結果を示します。

また論理演算の命令として，

AND命令　（論理積をとる）
OR命令　（論理和をとる）
XOR命令　（排他的論理和をとる）

などが用意されています。

●その他の主な命令

マシン語にも，BASICのGOTO文やGOSUB文にあたる命令があります。

JP命令　（指定されたアドレスにジャンプする）
CALL命令　（指定されたアドレスのサブルーチンをコールする）

これらの命令は，たとえば，

```
JP      1234H
JP      Z, 5555H
CALL    NC, 8888H
```

というふうに使います。

最初のものは，アドレスの1234_Hに制御を移します。次のものは条件付きジャンプと呼ばれるもので，Zフラグが立っているとアドレス5555_Hへジャンプし，立っていないと何もしません。そして3番目のものは，キャリフラグが立っていないとアドレスの8888_Hをサブルーチンコールします。

さてコールすると“戻れ”の命令が必要です。これが，

RET命令

です。

また，スタックポインタがあるので，スタックを操作する命令があります。それが，

PUSH命令　(レジスタのデータをプッシュする)

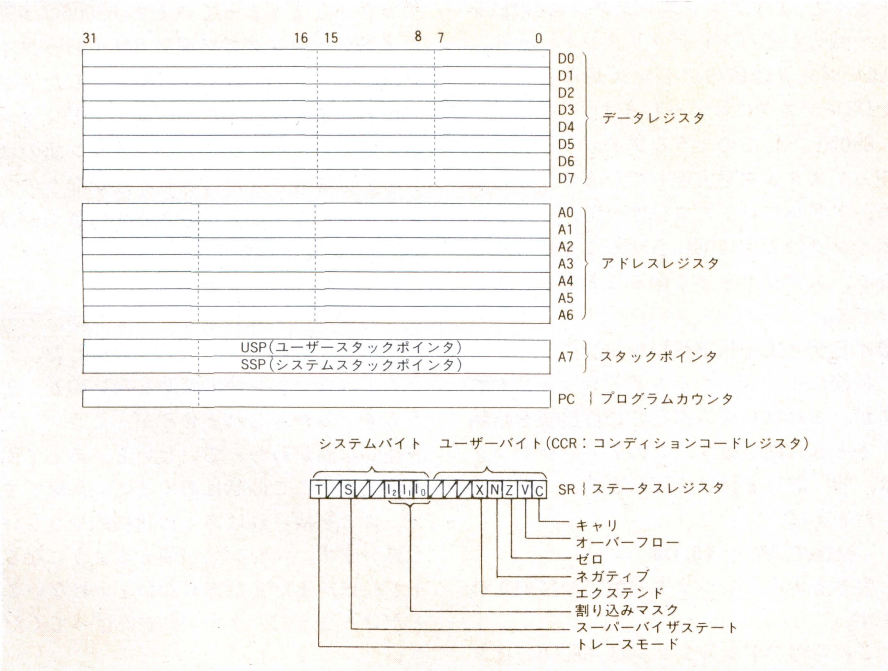
POP命令　(レジスタにデータをポップする)

です。

さらに，変わったところでは，データをビット列として扱い操作を施す**ローテイト命令**とか**シフト命令**などもあります。

さて，Z80の基本的な命令セットについてのみ説明してきましたが，ほかにも多くの命令があります。これらについては各自で勉強してほしいと思います。コンピュータの内部がちゃんと理解できていれば，決して難しくはないでしょう。大事なことは命令を覚えることではなく，いかにうまく組み合わせるかということです。

図13　68000のレジスタ構成



68000とマシン語

68000は68系の代表格といえるCPUで，Z80とは異なり16ビットCPUです。しかしその構造（アーキテクチャ）は極めて単純で，美しいものとなっています。

68000のレジスタ構成

まず，図13を見てください。これが68000のレジスタの構成です。はっきりいってZ80と比べると，すっきりしているといった感じがします。では，具体的に各レジスタの働きについて説明しましょう。

D0からD7のレジスタは32ビット幅の汎用レジスタで，まとめてデータレジスタと呼んでいます。しかし，単に汎用というだけではありません。なんとすべてがアキュムレータとして使うことが可能なレジスタなのです。したがって，データレジスタはすべて同じように扱えるわけです。

次に，A0からA7のレジスタはアドレスを扱うための32ビット幅のレジスタで，アドレスレジスタと呼ばれています。これも単にアドレスを格納しておくだけのものではなく，インデックスレジスタ，スタックポインタとしても使うことができます。したがって，たくさんのスタックを一度に使うことだってできるわけです。

さて，A7レジスタだけがちょっと違う構成になっています。というのは，このA7レジスタはスタックポインタとして使うことが前提となっているためです。といってもスタックポインタ以外の使い方も不可能ではありません。

68000は，走っているときに2つのモードを持っています。それが**ユーザーモード**と**スーパーバイザモード**です。そして，ユーザーモード時は**ユーザースタックポインタ**というアドレスレジスタがA7レジスタとなり，スーパーバイザモード時は**システムスタックポインタ**というアドレスレジスタがA7レジスタとなっているのです。普通私たちがマシン語を使うときは，ユーザーモードなのでA7レジスタはユーザースタックポインタとなっています。

残るPCレジスタ（プログラムカウンタ）とSRレジスタ（ステータスレジスタ）については「CPUの動作とレジスタの機能」で説明したとおりです。

68000ではバイト（8ビット）以外のデータ単位として，16ビットをワード，32ビットを**ロングワード**（Long Word）と呼んでいます。そこでレジスタの32ビット全部を使うか，16ビットだけ使うか，8ビットだけ使うかということを命令の後ろに，“.L”(ロングワード)，“.W”(ワード)，“.B”(バイト）を付けて示します。

アドレッシングモード

では，アドレッシングについて話を進めましょう。68000には，命令の対象となる数値，レジスタ，アドレスの扱い方に大きな特徴をもたらす，豊富な**アドレッシングモード**があります。このアドレッシングモードを生かすことで，プログラミングステップを簡略化できるばかりでなく，実行速度の向上が望めるのです。

68000のアドレッシングは，表1に示すとおり，6つの基本形に分けられ合計14種類のモードがあります。

ここでは，大まかに代表的な例を紹介しておきましょう。

●レジスタ直接アドレッシング

まずレジスタ直接アドレッシングですが，これは名前が凄いわりにたいしたことがないアドレッシングで，直接レジスタの名前をあげることでそのレジスタに格納された値を取り出して使います。

たとえば，

```
MOVE.W    D1,D3
```

と書くと，データレジスタD1の下位16ビットの値をD3にコピーします。なお，MOVE

命令はデータのコピーを行う命令です（Z80ならLD命令に相当します）。

●絶対アドレッシング

次に絶対アドレッシングですが，これは具体的にアドレスを値として示すというものです。

たとえば，

MOVE.L　D1,$1000

と書くと，データレジスタD1の32ビットをアドレス1000Hから1003Hの4バイトにコピーを行う命令です。なお68000の場合，数値が16進数だと“$”を頭に付けるので注意してください。

●PC相対アドレッシング

相対アドレッシングは，プログラムカウンタ（PCレジスタ）の値に対して相対的な値をアドレスとして使うものです。これはリロケータブルなプログラムを書くために使うものです。リロケータブルとはなにかということになるのですが，これについてはけっこう難しいのでここでは触れません。しかし，68000のアセンブラ（マシン語のプログラムを作るためのツール）は，できる限りリロケータブルになるようにがんばってくれますので，最初は気にしなくていいでしょう。

●アドレスレジスタ間接

アドレスレジスタ間接アドレッシングは，アドレスレジスタに格納された値をアドレスとするものです。

例としては，

MOVE.B　D0,(A1)

と書くと，レジスタD0の値をアドレスレジスタA1で示されるアドレスへ，コピーします。

●ディスプレースメント付きレジスタ間接

ディスプレースメント付きレジスタ間接アドレッシングは，上のアドレスレジスタ間接アドレッシングを拡張したようなもので，通常のインデックスレジスタの使い方と同じです。

例としては，

MOVE.B　D2,10(A1)

と書くと，レジスタD2の値をアドレスレジスタA1で示されるアドレスに10を加えたアドレスにコピーします。

●ディスプレースメント/インデックス付きレジスタ間接

これはディスプレースメント付きレジスタ間接を，もっと拡張したようなもので，インデックスレジスタを2つ指定するものと思えばよいでしょう。

例としては，

MOVE.B　D3,100(A0,A1.L)

と書くと，A0の値，A1の値とディスプレースメントである100を加えた値をアドレスとして，ここにD3レジスタの値を格納します。

これらの間接アドレッシングはインデックスレジスタの使い方を大幅に拡張したもので，うまく使えれば極めて強力です。

●プレデクリメント/ポストインクリメント付きレジスタ間接

プレデクリメント/ポストインクリメント付き間接レジスタアドレッシングは，いままでの間接アドレスアドレッシングをやる前や行ったあとに，アドレッシングに使ったアドレスレジスタの値を変化させるものです。

例としては，

MOVE.B　−(A0),D3

と書くと，まずアドレスレジスタA0の値を1つ減らして（プレデクリメント）から，A0レジスタの値のアドレスから1バイトをD3レジスタにコピーします。

68000では，このようなアドレッシングを使ってスタックを実現しています。このため，アドレスレジスタはすべて，スタックポインタのように動作させることが可能となり，スタックを多く作ることができるわけです。

●イミディエイトアドレッシング

最後にイミディエイトアドレッシングですが，これはレジスタなどに直接値を格納するときに使います。このアドレッシングは，“#”記号を付けて示します。

たとえば，

MOVE.W　#5,D0

と書きますと，レジスタD0に5が格納されます。

これらのアドレッシングは，68000のほとんどの命令で使えますので，よく覚えておいてください。

表1　アドレッシングモード

基本型	モード
レジスタ直接	データレジスタ直接 アドレスレジスタ直接
レジスタ間接	アドレスレジスタ間接 ポストインクリメント プリデクリメント ディスプレースメント付き インデックスおよびディスプレースメント付き
アブソリュート（絶対）	アブソリュート・ショート アブソリュート・ロング
プログラム・カウンタ相対	ディスプレースメント付き インデックスおよびディスプレースメント付き
イミディエイト	イミディエイト クイック・イミディエイト
インプライド（合意）	インプライド・レジスタ

68000の命令

それでは，マシン語の命令セットについて軽く触れて，68000については終わりたいと思います。まずデータの転送（コピー）を行う命令は先ほど述べたように，

MOVE命令（データのコピーを行う）

があります。次に算術演算命令として，

ADD命令　（加算を行う）
SUB命令　（減算を行う）
MUL命令　（掛け算を行う）
DIV命令　（割り算を行う）

などがあります。また論理演算を行う命令として，

AND命令　（論理積をとる）
OR命令　（論理和をとる）
EOR命令　（排他的論理和をとる）
NOT命令　（否定をとる）

などがあります。

これらのほかにZ80の場合と同様に，比較結果をステータスレジスタ（Z80ではフラグレジスタ）に反映させる，

CMP命令　（2つの値を比べる）

などやビット処理などのときに使う，ローテイト命令などがあります。

このほかプログラムのジャンプをする，

JMP命令

や，サブルーチンをコールする，

JSR命令

などがあります。

しかし全体的には，命令セットはかなりすっきりとまとまっています。68000は多彩なアドレッシングで処理をうまく行うタイプのCPUで，Z80のように細々とした決まりがありません。したがって，アドレッシングさえわかってしまえば，マシン語のほとんどが理解できたに等しく，かなりのプログラムを書くことができます。さすがは68000といえましょう。

＊　＊　＊

今回はマシン語の話というよりは，アーキテクチャの話となってしまいました。どうして直接マシン語の話をしないのかと思う方もいるかもしれませんが，こういった基礎的な話がわかっていないと，あとで困るのです。ここの話はあくまでも基礎ですが，ここを抜ければあとは比較的やさしいと思います。マシン語が使えるようになるまでには，まだまだ知らないといけないことがけっこうありますが，がんばってください。

アセンブラへの招待

Tan Akihiko 丹 明彦
Kuwano Masahiko 桒野 雅彦
Ogikubo Kei 荻窪 圭

私たちのまわりには，マシン語で書かれたソフトがあふれています。特に最近のゲームソフトの発達ぶりを見れば，マシン語の威力は説明するだけ無駄というものでしょう。でも，アセンブラは，そうした超高等ゲームなどのソフトウェアを作るようなプロだけのための道具ではありません。そうであってはいけないはずなのです。

「マシン語」と「アセンブラ」という言葉は無造作に使われることが多いようです。実際，この2つの言葉は，意識的に区別されることはあまりありません。「マシン語使える？」と「アセンブラ使える？」は，ほぼ同じ意味で使われています。

アセンブラとは何か

コンピュータが扱えるのは基本的に2進数だけです。ですが，私たちがそれを目にするときにはたいてい16進数に直されています。メモリの内容を16進で表示したものをダンプリストといいます。

このような16進数の数字列をメモリに書き込んでいけば，CPUはそれをプログラムとみなして実行していくわけですが，Oh!Xに掲載されているマシン語プログラムのリストを見ると，ダンプリストのほかにソースリストというものがありますね。CPUが扱えるのが2進数だけだとすると，これはいったい何を意味するのかという疑問を抱く人もいるでしょう。そうです。これがアセンブリ言語のプログラムなのです。

16進データが延々と続くダンプリストを見れば誰もが想像するとおり，人間が直接マシンコードを使ってプログラミングすることは特殊なケースを除いて事実上ほとんど不可能です。

マシン語の命令は1バイトから数バイトの単位で意味のある命令を構成しますが，意味のある命令である以上，もっと扱いやすい言葉に置き換えられないはずはありません。そこで登場するのがアセンブリ言語です。アセンブリ言語は，マシン語の命令をニーモニックと呼ばれる英語の省略形のようなものに置き換えたもので，マシン語の命令と1対1に対応します。

こうして命令をニーモニックで表してみると，

オペコード＋オペランド

という形をとっています。オペコードは動作を表す主命令で，オペランドは主命令の対象となるレジスタやメモリのアドレス，そして数値などです。

たとえば，Z80のニーモニックなら，

```
LD      A, 2BH
オペコード オペランド
```

といった具合です。これは，Aレジスタに16進データ$2B_H$をロード（LD）するという意味です。ちなみに，これをマシンコードに戻してみると，

3E 2B

となります。「LD　A」が3Eに対応していることに注意してください。これは完全に1対1の対応ですから，まったく同じことなのですが，1バイトの数値をオペコードとオペランド（の一部）に分解して表すことによりずっと理解しやすくなっているのです。

さて，もうお気づきだと思いますが，ニーモニックは人間には理解しやすくても，CPUには理解できません。コンピュータを動かすためには，ニーモニックを16進数（というか2進数）に変換してやらなくてはなりません。この変換作業をアセンブルといい，アセンブルを行うのためのツール（プログラム）のことをアセンブラというわけです。

アセンブラを利用した実際のマシン語プログラミングの大まかな流れは，次のようになります。

1） 設計

どういうプログラムを作るか，という構想を立て，アルゴリズムをじっくりと検討します。データ構造や，サブルーチンの構成などもこの段階で固めておきましょう。

2） コーディング

ソースプログラムを作成します。ここがプログラミングの中心であるように思われがちですが，本当は1)の設計がきちんとしていれば，2)は単純作業にすぎません。

3） アセンブル

リンカがあるシステムでは，リンクまで含む。コーディング作業の仕上げです。アセンブルエラーが出ないように祈りましょう。

4） デバッグ

エラーが出なくても，1発で動くプログラムなどまず存在しません。ここからが地獄の作業です。人によってはプログラミングの中で最高の楽しみだという人もいます。デバッガのお世話になるなどして頑張ってください。もう少しの辛抱です。

5） 完成

おめでとうございます。完成したプログラムで心ゆくまでお楽しみください。しかし，あなたに休息はありません。新たなる戦いの始まりです。

ご覧のように，アセンブラによるプログラミングのなかではリンカ，デバッガなどのツールも出てきますが，詳しくはあとのページで説明します。それでは，アセンブラの世界に入って行くことにしましょう。

図1　Z80のアセンブリストの例

アドレス	オブジェクト	ソースプログラム		コメント
8000		ORG	8000H	
8000	3A 00 90	LD	A,(9000H)	;9000H番地の値と
8003	21 01 90	LD	HL,9001H	;9001H番地の値を
8006	86	ADD	A,(HL)	;足して
8007	32 02 90	LD	(9002H),A	;9002H番地にしまって
800A	C9	RET		;終わる

プログラミング環境を知ろう

一般にアセンブラを使うといえば，マシン語プログラムの開発全般を指すといってよいでしょう。そもそもアセンブラとはどのようなものか，またそのプログラミング環境はどうなっているのでしょうか。

Tan Akihiko 丹 明彦

アセンブラへの招待

アセンブラを理解するには，まず対象となる機種のCPUをある程度知らなければなりません。マシン語を始めようと志す人は必ず，対象となるCPUの参考書を用意してください。X68000のアセンブラマニュアルにはニーモニックの解説が用意されていますが，アルファベット順配列という構成のため，学習には向いていません。また，実際のプログラミング時には，命令の一覧表のようなものを参照したほうがよいでしょう。

CPUとアセンブラ

CPUにはそれぞれ独自の機能や命令セットが用意されており，アセンブラの表記(ニーモニック）もCPUによって異なります。

Z80の命令は，

8/16ビットロード命令
交換，ブロック転送/サーチ命令
数値演算，論理演算命令
ローテート，シフト命令
ビット操作命令
ジャンプ，コール命令
入出力命令
CPU制御命令

に，68000では，

データ転送命令
数値，論理演算命令
ローテート，シフト命令
ビット操作命令
ジャンプ，コール命令
システムコントロール命令

にそれぞれ大別されています。

比べてみるとほとんど同じように見えますね。どんなCPUでも基本的な命令の種類はそれほど変わらないものです。

しかし，機能や使い勝手では，CPUによって大きく異なり，それが直接アセンブラの使いよさにかかわってきます。

まずは，アセンブラによるプログラミングの特徴について見ていきます。BASICなどの高級言語にはない概念がいくつかありますので，ここでポイントを押さえておきましょう。

●レジスタ

高級言語の「変数」にもっとも似ているのですが，同じものではありません。Cでいうポインタになることもあれば，通常変数と同じような扱いをされることもあります。なによりわかりやすい違いは，それぞれのCPUによって数が限定されているということです。数個から多くても数十個しかありません。

それから，当然ながら扱えるデータの種類は2進整数しかありません。命令によっては10進や16進の処理ができることもありますが，基本的には2進数としてしか扱いません。

●ラベル

マシン語の命令のなかには，どうしてもアドレスを指定しなければならないものが数多くあります。そんなとき，アドレスの代わりになるのがラベルです。アセンブルリスト中，ジャンプ先やデータエリアにつけておく名札のようなものです。BASICの行番号やラベルにあたります。

●条件分岐

BASICのIF〜THEN文にあたる処理ですが，たいがいのマイクロプロセッサは，データの比較と条件分岐を次のように行います。

1) 比較するデータ間で適切な演算を行う。比較演算は，一般には引き算を仮に行って(引き算の値はどこにも出てこない)，その結果から次の処理をする。

2) 演算結果はフラグに反映されるので，条件ジャンプで分岐する。

つまり，BASICでは，

IF A>B THEN GOTO〜

と1文ですませていたものも，アセンブラでは，

・A－Bを行う
・フラグが「演算結果が正」を示していたら，〜のアドレスに飛ぶ

というふうに2段階に分けなくてはなりません。しかも，THENの後ろはジャンプかコールくらいしか許されず，そのほかの処理をしたいなら，別の書き方を工夫しなくてはなりません。

たとえば(ちょっとBASIC的な書き方をしますがお許しを)，

IF A>B THEN PRINT A

は，

```
    IF A<=B GOTO LABEL
    PRINT A
  LABEL:
    ⋮
```

のように，ジャンプ命令でスキップする形でしか書けません。

フラグは，演算結果を反映するものだといいましたが，その正体は数ビットのレジスタで，その各ビットが演算結果のいろいろな性質を表すのです。演算結果が負になったとか，0になったとか，みんなこのフラグレジスタに入っています。「フラグ」とは「旗」のことです。演算結果によってフラグレジスタのビットが立つ(“1”になる）ことを「フラグが立つ」といいます。演算結果が0になったときは，ゼロフラグという旗が立ってCPUに「0になったよ」と知らせてくれるのです。

フラグの概念は高級言語には馴染みの薄いものです。また，「比較≒演算」という考え方もBASICに慣れた人には少し難しいかもしれません。Cになると，このあたりが露骨にアセンブラ的で，比較も代入も，さらに広い意味ではサブルーチン（関数)

思い出のブロック転送

私は昔，スクロールゲームが全盛の頃，画面スクロールの技術を知りたくて，あるプログラムを解析したことがあります。そのとき使ったのが，ちょっと変わった逆アセンブラで，命令をBASIC風に解説してくれるというものでした。たとえば，ブロック転送のLDIRなら，

```
$XXXX   POKE DE, PEEK (HL) :
        HL=HL+1:DE=DE+1:
        BC=BC-1:
        IF BC=0 THEN GOTO $XXXX
```

といったぐあいにです。私はこうしてブロック転送を理解しました。

LDIRは，私が最初に覚えた，多少複雑なZ80マシン語の命令だったのです。

さえも，みんな演算の仲間で，値を持っています。

●シフト，ローテート，ビットテスト/セット/リセット

これはいよいよアセンブラの非高級言語的部分です。高級言語では，数値データをビット単位でのぞいたり，内部表現を調べたりすることには積極的ではありません。これは特にグラフィックを扱うときに大きな弱点となっています。

それがマシン語ではどうかといえば，レジスタは2進整数で，しかも長さは決まっています。ときにはレジスタの値が，数値でなく数個の別の意味を持ったビットの集まりを表している場合もあります。シフト命令などは，まさにマシン語のこうした性質を反映した命令といえるでしょう。

レジスタのビットをズラしたり，特定のビットだけをいじったりすること自体は，数学的に見ても意味がないことのように思えますが，特定の性質を持ったデータを処理するためには有効です。

たとえば，X1などのグラフィック画面は，横に8つ並んだドットで1バイトを構成し，それぞれのビットはお互いにまったく関連がありません。先ほど出てきたフラグレジスタも，お互いのビットの間には少しも関連がなく，したがってフラグレジスタの値自体は少しも数学的意味を持たないのです。このように高級言語はこの手のデータの取り扱いを苦手とします。マシン語が威力を発揮する分野といえるでしょう。

●変数のアクセス

くどいようですが，マシン語では基本的にレジスタと，レジスタを使って直接にアクセスできるデータに対して，ひとつの命令でひとつずつしか処理できません。それにアドレスの概念は，BASICにはないものだけに，アセンブラ入門のためには避けて通るわけにはいきません。

68000にはアドレスレジスタとデータレジスタがはっきり区別されています。Z8000のように，両者は区別しないという思想もありますが，C言語の経験者ならわかるように，ポインタとデータを混同すると，惨劇が待っているのです。7月号のC言語特集でポインタがよく理解できなかった人は，アセンブラで理解してほしいと思います。

マシン語がデータをレジスタでしか扱えないことの証拠として，BASICの代入文，

X1＝X2＋X3

をハンドコンパイル（！）してみましょう。当然変数X1, X2, X3の値はメモリのどこかに格納する必要があります。

```
Z80
     LD A, (X2)
     LD HL,X3
     ADD A, (HL)
     LD  (X1),A
      ⋮
X1: DS 1
X2: DS 1
X3: DS 1

68000
     move.w   X2,D0
     add.w    X3,D0
     move.w   D0,X1
      ⋮
X1: ds.w    1
X2: ds.w    1
X3: ds.w    1
```

どちらも，メモリ上のX2の値を格納しているアドレスからX2の値をレジスタにロードし，それにX3の値を足し，それをX1の値を格納すべきアドレスにロードする，という順番をとっています。

ラベルX1, X2, X3は，変数を格納するポインタとしての働きしかなく，高級言語のように，直接的に変数の値としてレジスタと同じように働くということはありません。C言語の文字列は，これに少しだけ似ています。

簡単なところで，「番地ADRS1から格納されている長さLENGTHのデータを番地ADRS2以降に転送する」プログラムです。Z80と68000の両方で書いてみましょう。

```
Z80
        LD HL,ADRS1
        LD DE,ADRS2
        LD BC,LENGTH
        LDIR
        END
68000
        lea.l      ADRS1,A1
        lea.l      ADRS2,A2
        move.w     #LENGTH－1,D1
loop:   move.b     (A1)＋, (A2)＋
        dbra.w     D1,loop
        end
```

この2つのリストを比べると，Z80と68000の設計思想（ちょっと大げさですか）の違いが見えるような気がします。

Z80は，内蔵された個々のレジスタの機能や用途を専門化しています（たとえば，LDIRの場合，BCが暗黙のうちにカウンタとして使用されていることなどです。そのほかの命令でもBCレジスタは，カウンタとして働くことが多いのです）。そのため，1つひとつの命令はシンプルにまとまっています。ただし，その副作用として，動作モードが完全に1種類しかなくなっています。レジスタの使い方が1通りしかないため，使い回しに少々頭をヒネることになるわけです。

対して68000は，転送元，転送先，カウンタにいちいちレジスタの番号を指定していたり，ループをわざわざ作っていたりする点など，少々単純さを欠いているように見えます。これは，アドレッシングモードを充実させた副作用のようなものでしょう。

リロケータブルアセンブラ

アセンブラは，出力するオブジェクトのタイプによって2つの種類に分けられます。すなわち，アブソリュートアセンブラとリロケータブルアセンブラです。

アブソリュートアセンブラとは，プログラムの先頭アドレスが固定されたオブジェクトを生成するものです。このタイプのアセンブラでは，あらかじめ配置アドレスを決めておき，プログラミングの際にはソースリストのなかでそのアドレスを指定しておかなくてはなりません。

一方のリロケータブルアセンブラの場合は，リロケート（再配置）可能なオブジェクトを生成し，どのアドレスにオブジェクトを置いても実行できるというものです。

リロケータブルアセンブラのほうが利用する際の都合で配置アドレスを変更することができるので便利なのですが，Z80のマシン語では，完全にリロケータブルなオブジェクトを作ることは困難です。というのも，もともとCPUの命令体系がそのようにできていないのです。Z80用アセンブラの多くがアブソリュートアセンブラなのはそのためです。ただ，Z80でもちょっとしたトリックを使えば，リロケータブルなオブジェクトが作れないわけではありません。

そのためには，オブジェクトの中でアドレスにかかわる命令が入っている箇所をチェックしておき，それをリロケート（再配置）情報として持っておく方法があります。こうすると，実行時にアドレス関係の部分だけ書き直してメモリ上に配置するように仕組んでおくことができ，疑似的にですがリロケータブルなプログラムを実現できます。この作業のために専用のプログラムが用意されるのが普通です。

68000ならリロケータブルなオブジェクトを作れるので，リロケートのための特別

なプログラムは必要ありませんが，処理速度はアブソリュートプログラムのほうが速いので，Z80と同じような方法をとることもあります。

分割アセンブルとリンク

アセンブラによるプログラミングにおいてはリロケータブルかどうかよりも，もっと重要なことがあります。処理内容が複雑であればあるほど，プログラムサイズは大きくなり，サブルーチン呼び出しなども頻繁になります。アセンブラ本体やソースプログラムを格納するメモリ領域も馬鹿になりません。特にメインメモリがたくさん取れない場合にアセンブル可能なプログラムの大きさが制限を受けてしまうこともあります。

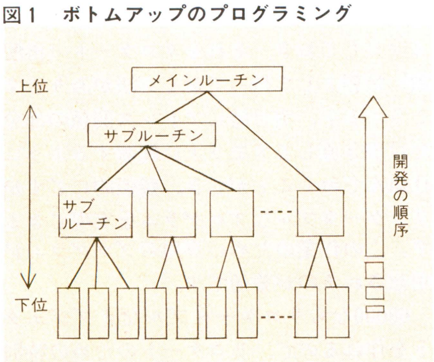
そこでクローズアップされてくるのが，プログラムを分割するという手法です。コンパイラでもこのような手法は多く用いられ，特にCコンパイラでは，ライブラリの充実度が大きなポイントになっています。

分割アセンブルによって得られるメリットは，まずさっきもいったように，作りたいプログラムの大きさがメモリ容量のために制限されずにすむということです。また，きちんと動くようになったサブルーチン群を分離していけば，いちいち実行のたびに全部のルーチンをアセンブルし直す必要がなくなります。

さて，いくつかに分けて別々にアセンブルしたプログラム群を最後にひとつのオブジェクトにまとめ上げることをリンクといいます。当然，リンクのためのツールをリンカと呼ぶわけですが，リンカを利用するのはリロケータブルアセンブラの場合で，アセンブルされたオブジェクトはリンカを通すことで初めて実行可能な形式のオブジェクトとなります。

まず，リロケータブルアセンブラでこの分割アセンブルの手法を使えば，昔作ったプログラムのサブルーチンをそっくり利用したいという場合にも活用できます。いちいち昔のプログラムのソースリストから欲しいサブルーチンだけ切り取ってきていま作っているソースプログラムのなかに埋め込んだり，1から作り直して無用のバグを出したりせずにすむのです。必然的にメモリと時間が節約できることになります。

図1　ボトムアップのプログラミング



サブルーチンは，ひとつ作るたびに動作チェックを行い，最後に作るメインルーチンで各サブルーチン間の連絡や調整をやるのが常道です。これを，ボトムアップのプログラミングといいます(図1を見てください)。極端な例では，メインルーチンにはループのなかに10個余りのCALL文だけが並んでいるようなプログラムにも出会ったことがあります。処理はみんな下位サブルーチンに下請けです。こういうプログラムを作るときにこそ，リンカの働きは大きいといえます。

一方，アブソリュートアセンブラの場合の分割アセンブルはもう少し単純で，もともと絶対アドレス指定のオブジェクトを生成するため，特にリンカのお世話になることはありません。作業は簡略化しますが，そのかわり，配置アドレスの重なった複数のプログラムを利用したいと思うとかなり面倒なことになります。

環境に応じた選択を

このように，一般にリロケータブルアセンブラを使って「分割アセンブル→リンク」という開発手順を踏むことが開発効率からいって好ましいのは確かなのですが，これはCPUやそのマシンの開発環境，そしてユーザーのレベルなどによって事情が変わってきます。

たとえば，X68000の場合，オマケとしてアセンブラAS.XとリンカLK.Xが福袋というディレクトリに入っていました。ACEからはTHE福袋V2.0として別売になりましたが，アセンブラマニュアルとプログラマーズマニュアル（Cコンパイラに付いているのと同じもの）が付いて，より安心して利用できる状況になりました。

このX68000用のアセンブラ（とリンカ）は，複雑なプログラムや大規模なプログラムの開発には向いています。しかし小さなプログラムを作るのにも必ずリンカを通さなくてはならなくなっていて，開発時間的にはかえって損をすることもあります。数個のオブジェクトファイルをリンクして実行ファイルを作っているのですが，オブジェクトファイル間でサブルーチン呼び出しやデータワークエリア参照をしあっているのが普通なので，ラベルのチェックなどにかなり手間取ります。アセンブルの際にエラーが出なくても，リンクのときにラベルがありませんというエラーを出されてムッとしたことのある人も多いと思います。しかも，アセンブル行程のなかで一番時間のかかるのは，実はリンクで，RAMディスクを使ってさえ分単位の時間待たされることもざらにあります。

Z80の場合などは特にそうですが，もしあなたがマシン語の初心者で，高速な記憶装置を持っていないなら，ぜひともオンメモリですむエディタアセンブラをお勧めします。本誌で発表したZEDAや今回のREDAがあれば十分でしょう。エディタアセンブラなら，初めにシステムをロードしておけば，あとはソースプログラムの修正もオブジェクトプログラムの生成も思いのままです。もしオンメモリで収まるようなプログラムなら，エディタアセンブラもソースファイルも，それにオブジェクトファイルも一度に扱えます。

ちょっと贅沢なデバッガ

さて，次はデバッガのお話に入りましょう。プログラム中に潜む誤りのことをバグといいますね。そしてバグを取り除くことをデバッグといいます。

デバッガといっても，デバッグをやってくれる，つまりプログラムのバグを全部潰してくれるというソフトウェアではありません。デバッグをするのはあくまでプログラマ自身，デバッガはそのお手伝いをするだけです。虫取り網だって，虫をひとりでに捕まえてはくれませんしね。

一般的なデバッガの持っている機能に触れてみましょう。表に概略をまとめておきました。デバッガというからにはバグが出たときに役立つものでなくてはなりませんが，ひと口にバグが出たといっても，その症状はさまざまです。

バグの原因が画面にモロに現れるようなバグだったら対策も立てやすいでしょうが，アセンブラの場合，無礼にもプログラマに断りもなく暴走してしまうことのほうがむしろ多いのです。そうしたときの手掛かりはどこに求めたらいいのでしょうか。カンに頼ってソースリストの上をあちこち捜し回るのもいいかもしれませんが，それよりもコンピュータ内部でどういう処理がされているかを組織的に調べるのが早そうです。

●メモリチェックとレジスタ設定

プログラムを少し実行しては止め，そのたびにワークエリアなどのメモリ内容やレジスタの値をチェックするのが効率よさそうです。そのために，デバッガにはメモリチェック機能や，レジスタダンプ機能が用意されていて，メモリ内容やレジスタの値を見たり変えたりできます。ワークエリアやレジスタの値を設定してからプログラムの実行を再開すれば，思いのままの動作条件でプログラムが実行できます。

●ブレイクポイント

ここで，ちょっと待ってください。マシン語のプログラムの実行を「止め」，「再開する」にはどうしたらいいのでしょう。BASICのプログラムと違って，マシン語の場合は「SHIFT＋BREAK」なんて重宝なものは効きません。なにかいい手が欲しいところです。

というわけで，デバッガにはブレイクポイントを設定する機能が付けられることになります。デバッガを起動し，ターゲットプログラムの数箇所にブレイクポイントを仕掛けます。具体的には，ブレイクポイント処理ルーチンにジャンプするような短い命令コードを書き込むようです。

Z80トレーサなら，リスタート(RST)命令が1バイトなのでそれを使っています。その飛び先（書き込んだ命令がたとえばRST38Hなら，0038_H番地）からブレイクポイント処理ルーチンが走ることになります。

ブレイクポイントの設定が終わったら，ターゲットプログラムの適当な番地へジャンプします。実行が進んでブレイクポイントまでやってくると，デバッガのほうでストップをかけてくれるのです。もうデバッガのコマンドレベルに戻っているのですから，レジスタダンプなり，メモリダンプなり，好きにできます。さらに，ループを作ってあるプログラムで，1回1回ブレイクポイントを通るときに中断されると面倒な場合などは，「～回目までは素通りして，その次の回で実行を中断する」ようなブレイクポイントを設定できる器用なデバッガもあります。

●トレース

もうひとつ，先ほどトレーサという言葉が出てきましたが，デバッガにはトレースという機能が装備されています。これは1命令実行するたびに実行を停止させる機能で，1ステップごとに実行の過程が追えます。ただし，トレーサを使うとスピードがガックリと落ちるので，たとえば初期設定を行う部分が巨大なものだったときなど，死ぬほど退屈させられます。こういうときは，その初期設定をしている部分にバグがないことを確認したうえで，ブレイクポイントをそのあとに設定し，初期設定が終わり，ブレイクポイントで止められた時点からトレーサを働かせるのが賢いやり方です。

そうして，サブルーチンごとにチェックをかけていき，プログラムのデバッグをすることができるわけです。このように，デバッガはマシン語プログラムの開発にはたいへん便利なものです。

●逆アセンブル

最後に逆アセンブル機能です。ターゲットプログラムが自分で作ったプログラムで，ソースリストが手元にあるならまだいいのですが，システムプログラムを解析したいときなどには，ソースリストが手に入らないのが普通でしょうし，ソースリストを持っていても，画面に出てくる16進の命令コードの羅列と，ソースリストのアセンブリ語とを対応させるという苦しい作業もやってられません。そこで，16進コードをアセンブリ語に戻してくれる逆アセンブル機能が登場するのです。

逆アセンブルの手順

X68000のDB.Xを使ってなにか逆アセンブルしてみましょう。なんでもいいのですが，ディレクトリBINにあるツールのうちでもっとも小さいSYS.Xにします。

```
DB SYS.X
```

のようにデバッガからプログラムを読み込み，起動します。ここで，

```
-P
```

と入力してプログラムの読み込まれているアドレスを確認します。開始番地と終了番地が表示されますのでそれを記録しておき，

```
-Q
```

でいったんデバッガを抜けます。次に，

```
DB SYS.X > SYS.S
```

のように，画面出力をファイルにリダイレクトするようにデバッガを起動します。画面にはなにも出てきませんが，かまわず，

```
L　開始アドレス　終了アドレス
Q
```

と入力します。

アセンブリ語とマシン語のコードは，必ず1対1に対応しているので，逆アセンブラで出力したアセンブリ語は，ソースリストのそれと一致します。ここで作った逆アセンブルファイルを加工していきさえすればソースプログラムを得ることもできるのです。

表1　デバッガの主な機能

メモリチェック	メモリダンプ： お馴染みの16進数の羅列 メモリチェンジ： メモリ内容の書き換え
レジスタ設定	レジスタやフラグの値を見る 値を設定することもできる
ブレイクポイント	好きなところで実行を中断する
トレーサ	1命令ずつ実行してプログラムの流れを追う
逆アセンブラ	マシン語プログラムをアセンブリ言語に逆変換する

ただし，デバッガではラベルや注釈文などは復活しません。それから，データエリアのデータも復活しません。逆アセンブラのなかには，データエリアを指定しておくとそこだけ別に処理してくれるものもありますが，そうでなければ，逆アセンブラはプログラムとデータの区別が少しもできません。どちらも逆アセンブラにとっては16進数のコードの羅列にすぎないから当然といえば当然ですが，このせいで文字列などは思いもよらない命令にバケてしまいます。

デバッガを使ううえで気をつけなくてはならないのは，ブレイクポイントの設定や逆アセンブルなどは，きちんと命令の区切りのところで行うということです。1命令が1バイト（Z80の場合。68000なら1ワード）ですむものならいいのですが，2～4バイトになるときのほうが多く，その途中を切れ目と間違えるとえらい目にあいます。そこのところだけ（ときにはかなり後ろまで）全然，別の命令になるので，当然のことながらなにを処理しているのかわからなくなってしまいます。

デバッガは使い慣れると強力な味方になりますが，どうしても必要なものではありません。実力も実績もあるプログラマでも，デバッガは使っていないという例はちらほら見かけます。要は，自分に合った開発環境なのです。

＊　＊　＊

マシン語プログラムの開発は決して楽なものではありません。デバッグで行き詰まったときは，もういやだ，マシン語なんてもう二度と付き合いたくない，と一度は思うことでしょう。しかし，それが完成してしばらくすると，また作りたいと思うようになるのです。さあ，あなたもどんどんハマってください。素敵な世界があなたを待っています。

かしこい孫の手の使い方

アセンブラにはアセンブラの利点，高級言語には高級言語の利点というものがあります。これらをうまく使い分けるのが賢いプログラミングスタイルです。ここではマシン語の「使いどころ」を考えてみましょう。

Kuwano Masahiko 桒野 雅彦

アセンブラへの招待

痒いところに手が届く

最近はCが主流になってきていて，以前はBASICからアセンブラへという流れだったのが，最近ではいきなりCという方も少なくないでしょう。私も最近はCばかり使っています。

特にX68000になると，XCがあれば細ごましたアセンブラ的な処理からBASIC並みのことまで簡単にできてしまうのです。以前のように，BASICかマシン語かという選択しかなかったころに比べれば，アセンブラの登場を願う場面がかなり少なくなっているのは間違いありません。

しかし，それでもアセンブラが不要とはいえません。まだまだ，細ごましとした処理ではアセンブラに勝る言語はありません。一般にアセンブラが使われるのは，次のような理由である場合が多いようです。

1) **オブジェクトがコンパクトで速い。**
2) **さまざまなバイト数のデータが混在しているような場合には高級言語よりもすっきりと書ける場合が多い。**
3) **ビット操作（シフト／ローテートを含む）関係ではアセンブラのほうが強力な場合が多い。**
4) **CPUやハードウェアに密着した部分ではスタックの切り換えやタイミングに厳しい部分のアクセスを伴ったりするため高級言語が使えない。**

1)については低レベル高級言語とも呼べるCの普及で，若干色あせたとはいうものの，2)以降についてはまだまだアセンブラのほうが便利，あるいはアセンブラでないと実用に耐えないような場合が多いでしょう。BASICやCで行き詰まってくるのもこのような場合であると思います。

そんなとき，その部分だけをちょこちょこっとアセンブラで組んで外部関数などにしてやることで，ずっとエレガントに解決できる場合が多いのです。高級言語では手の届きにくい，痒いところにアセンブラという孫の手を貸してあげることで，ぐっと気の利いたものになるというわけです。私は8086系のCPUもよく使うのですが，そこでも作るプログラムのほとんどすべてがCとアセンブラの組み合わせです。もっとも8086系でC（インテルが出しているCであっても）を使うと，痒いところが多過ぎるために孫の手だらけの千手観音のようになってしまいますが。

さて，アセンブラのこのような使い方は，マシン語を勉強しようとする人にとってもよいことです。高級言語にエッセンスとしてアセンブラを使い，そこで必要なパラメータを高級言語側から与えるようにするのです。すると，アセンブラの部分だけではサンプルプログラムの域を出られないようなものでも，高級言語側からの操作でかなり複雑なこともできるようになりますから，面白さが段違いです。

たとえば，画面の指定位置にドットを打つプログラムをアセンブラで組んだとすると，これを使ってディスクから画像データを読み込んで表示させることも，複雑な関数をグラフ化することもできるのですから。すべてアセンブラで頑張ろうとしたらこうはいきません。

高級言語との住み分け

アセンブラは低水準言語といわれるように，命令の数もそれほど多くはありませんし，ごく基本的なものばかりで，それ自体は決して「難しい」ものではありません。難しいといわれるのは，BASICインタプリタやCで書いていることと同じことをいきなりやろうとしてしまっている場合が多いようです。そもそも，高級言語というものはアセンブラだけではプログラミングが大変であるからということで作られたのですから，それらの得意分野で勝負しては手間がかかってしかたがないのが当たり前です。

高級言語として，BASICやCを取り上げたとき，これらがアセンブラと比べて圧倒的に優れているのは，

1) **制御構造の記述**
2) **変数の取り扱い**
3) **数値計算（特に浮動小数点）**

などでしょう。これらをアセンブラで書くのはやや面倒な部類に入ります。たとえば1)の制御構造の記述についていえば，アセンブラではwhileはおろかif～then～else～といったものすらありません。分岐についても，BASIC風に書くならgoto, gosub, if～gotoの3種類のごく原始的なものしかありません。

これだけを見ても，アセンブラで複雑な構造のプログラムを組むのは「難しくはないが面倒臭く，間違えやすい」というのが感じられるでしょう。このようなところにいきなり挑戦するのはやめましょう，ということで今回は高級言語とアセンブラの「住み分け」という考えを持ち込もうというわけです。

実際のプログラミングに関しては，78ページからの「初めは誰でも文字表示」を見ていただくとして，ここでは基本的な考え方を中心にお話ししましょう。

住み分けの基準はさしあたり次のようにしておきます。

1) **ワークエリアは極力，レジスタだけで済むようにする。当然，演算もレジスタ間だけで済むようにする。**
2) **分岐は3つ以下(ループは除く)。**
3) **ひとつのサブルーチンの長さは50行以下。**
4) **これらにあてはまらないときはサブルーチンを分割したり，高級言語に処理の一部を移す。**

これだけ制約をつけてしまうと本格的なプログラムを組むのは難しくなりますが，アセンブラに慣れるまではこのくらいの制約をつけて頭が混乱しないようにしておいて，慣れるに従って次第に制約を外していくようにするとよいでしょう。

ハンドコンパイルリスト?

実際には，これだけ制約をつけてもなお，アセンブラでプログラムを作るのは面倒なものです。特に，アセンブラではリストが1次元的にずらずらと並んでおり，1つひとつの命令の動作が細かいので見づらいも

のです。これがアセンブラの扱いにくさのひとつの要因でもあるでしょう。

そこで，自分自身のためにもプログラムの横にはコメントをつけましょう，となるのですが，気の利いたコメントを考えるというのはなかなか難しいものです。「ジングリッシュ」ともいわれるようなあやしげな英語を使ったりすると他人はおろか自分でもよくわからなくなりますし，といって日本語を使おうとすれば，こんどはフロントプロセッサのON/OFFを繰り返すことになって，面倒きわまりないのです。

私自身もこのような事態に苦しまされてきました。すったもんだとしているうちにたどりついたのが，コメントをCなどのソースで書いておく方法で，勝手に「ハンドコンパイル」と命名しています（本当は逆コンパイルなのだが)。完成したソースリストを見ると，横にあるCのソースがコンパイルされて左のアセンブラのソースになったように見えるので，このように呼んでみました。図1に示すのは，こうして作成したソースリストの例で，今回の特集の入門用に作った文字を表示するプログラムの一部です。

Cのソースですから当然，フロントプロセッサのお世話になる必要はありませんし，やれ主語はどれだ，動詞がない，ありゃ動詞が2つじゃ同士討ちじゃないか，などと悩むこともありません。特に難しい処理をしているところだけ日本語のコメントを付加しておけば，それで必要かつ十分なのです。特に16ビットCPUではCとアセンブラのギャップが小さいので，かなり簡潔に表現できますし，アセンブラのどのへんでどのような条件判断やフラグ操作をしているかが，コメントを「読む」のではなく「眺める」だけで判断できるという大きな利点があります。

図1　ハンドコンパイルリスト？

コンパイルされたように見える

アセンブリ言語によるソース ← C言語によるコメント

```
                                              * int    char_data,x,y;
                                              * short i,j;
character_output:                             * character_output()
                                              * {
            bsr      init_screen              *     init_screen();
            move.l   #$000c3021,d1            *     char_data = 0xc3021;
            moveq.l #100,d2                   *     y = 100;
            moveq.l #9,d4                     *     i = 9;
    char_out_loopy:                           *     do {
            move.l   #200,d3                  *        x = 200;
            moveq.l #8,d5                     *        j = 8;
    char_out_loopx:                           *        do {
            bsr      print_character          *               print_charcter();
            addq.l   #1,d1                    *               char_data++;
            add.l    #40,d3                   *               x += 40;
            dbra     d5,char_out_loopx        *           } while (j-- > 0);
            add.l    #30,d2                   *        y += 30;
            dbra     d4,char_out_loopy        *     } while (i-- > 0);
            dc.w     _EXIT                    * }
```

ソフトウェアの階層構造

プログラムがハードウェアと一体化している組み込み型マイコンや，自作のハードウェアにプログラムを載せる場合などでは，なにをするにも1から10まですべての面倒をみなければいけませんから，LSIのレジスタ操作や，VRAMへの直接書き込みなど，非常にハードウェアに近いレベルでの操作が基本になります。TK-80を始めとするワンボードマイコンが主流であった時代には，これが当たり前のことでした。

これが，CRTやキーボードを始めとする周辺装置を抱え込んだパソコンになると，カセットテープのロード/セーブや画面表示のようなごく基本的な入出力動作についてはROMの中のサブルーチンとして用意されるようになってきました。これをさらに進めたのがX1turboやMZ-2500で，ハードウェアに密着した部分の基本的な操作や数値演算などルーチンをROMの中に集め，IOCS(Input Output Control System）としています。

IOCSには，モードコントロール，テキスト表示，カーソルコントロール，ジョイスティック，プリンタ，ディスク(FD/HD)，カレンダークロック，ADPCM，OPMなどの周辺デバイスの基本操作，それにグラフィック，PCG，スプライトのコントロールとほとんどのことはできるようになっています。

このIOCSの存在が，パーソナルコンピュータとしては珍しいソフトウェアの階層構造を作る鍵になっているのです。X1/MZではハードウェア寄りの処理をIOCS コールで行うことで，ソフトウェアをハードウェアから切り離しておけますから，たとえばOSを載せるときにはファイル管理などのよりハイレベルの入出力を IOCS レベルを使って実現するようにされるわけです。

OS上で動くCコンパイラなどはもちろん，DOSコールと呼ばれるOSのシステムコール（ファンクションコールともいう）を通して動きます。さらにCコンパイラの入出力ライブラリの記述は基本的には（行儀のよいものは）OSのシステムコールを使って行われます。もちろんC言語で作られた標準的なプログラムはすべての入出力処理をライブラリ関数を通して行います。

この関係は次のようにまとめられます。

アプリケーション
↓
ライブラリ関数コール
↓
OS(DOS)コール
↓
IOCSコール
↓
ハードウェアアクセス

上の層は下の層が実現していることを使って，さらに高度なことを実現していくわけです。ただし，「高度」というのは「自由度」との引き換えであることは覚えておいたほうがよいでしょう。上の層になるほど特定の使い方をする場合には非常に簡単に行えるようになる一方，そこからすこし外れると，とたんに難しくなったり，場合によっては不可能ということになるわけです。

このような事情もあって，各層がさらに分解されたり，自分の2つ以上下のレベルに直接アクセスすることもないわけではなく，たとえばXCのライブラリではIOCSコールを使うレベル0からBASICレベルであるレベル3までの4レベルに分類されています。ただしこのように階層を飛びこえてより下位の層に直接アクセスするということは，その層の上にある層のソフトウェアが管理しているところに割り込んでいくことになるわけですから，むやみに混在させて使うと，上の層のプログラムが混乱することになる場合があります。

XCのライブラリの説明書に「なるべく同一のレベルだけでプログラムを組むようにしてください」という，わかりにくいメッセージが書いてあるのはこのためです。

アセンブラでプログラムを作るときは，いわば「無敵モード」ですから，この層のどこを使おうと自由です。一般的に目的とする機能がいろいろな層で実現できる場合には，なるべく上の層を使うほうが楽ではありますが，その引き換えとして実行時間は余計にかかる傾向があります。楽を取るか速度や苦労（楽しみ？）を取るかの取り引きになるわけですね。

カウチポテトチップス・アンデリシャス・ゴールデン・アセンブラ・ブルース

アセンブラは誰にでも使えるツールですと言っては，初心者をその気にさせようというのが世のマシン語特集の常。ですが，逆にアセンブラのわかりにくさというものを本音で語ってみるのも一興でしょう。

Ogikubo Kei 荻窪 圭

アセンブラへの招待

私はいわゆるアセンブラには近寄らないようにしている。理由はまあ，いろいろ胸の内に秘められてはいるが，なんだかんだいって手を出さないように気をつけているというのが真実に近いところだ。

編集部の人はみなそのことを知っているはずなのに，なぜか，アセンブラの話を依頼されてしまった。困ったもんだ。私に原稿を頼んでしまった編集部の方に御霊前を捧げたい。私の夢はマシン語バリバリOSばりばりアーキテクチャごしごしアルゴリズムてこてこのエンジニアではなく，のほほ〜んとロッキングチェアに座って，マウスやらなんやらでへへらへらへらとグラフィックやら音楽やらゲームやら，そんでもってまだ見ぬ新ジャンルのソフトと戯れる電脳カウチポテト族になることなのである。

わがままな奴

マシン語はわがままである。まず，Z80やら68000の持つ命令だけ知っていても，なんの役にも立たないところがわがままである。それはそうだろう。CPUさんは，ディスプレイに線を引くどころか点ひとつ打つこともできない。できるのはI/Oやらメモリのアクセス，それから簡単な計算と，ビット演算程度のもの。CPUさんにすれば，わけもわからず命令通りに動いていると，遙か彼方のほうでなにやらディスクがガギゴギ叫んでいたりCRTがテラテラと光っていたりという状況だ。逆にいえば，人間がマシン語でCRTをテラテラさせるには，CRTからCPUへの長い道のりを辿っていかねばならないのだ。

確かに，アセンブラなら直接16進数のマシン語を操るよりは遙かに楽で，マシン語の代わりにニーモニックなる命令を使えば，あとはよきにはからってくれると思っていい。しかし，そんなくらいで私はごまかされはしない。私は誰がなんといおうと，アセンブラは嫌いだ。

理由1：アセンブラは金がかかる

まず，アセンブラを入手せねばならない。ここで考えるべきはOSである。たとえば，X68000ならHuman，98ならMS-DOSとなる。では，X1やMZはどうするか。2つにひとつをまず選ばねばならない。CP/MかS-OSか。

そしてCP/Mだと，シャープから出ているのが9,800円（とんでもなく安いほうだ）に加えて，マクロアセンブラMacro-80(2万円）あたりを買いたいところである。

S-OSにすると，金はかからないけれども，SWORDを打ち込み，アセンブラを打ち込み，エディタを打ち込み，などとやっていると，時間がかかったりして，時は金なりというくらいであるから，金がかかったのとあまり変わりはない。

で，アセンブラが手に入ったとして，まだまだお金はかかる。アセンブラに入門するには，少なくとも2冊の本が必要だからだ。1冊はアセンブラの入門書である。基本的なルーチンの例なんかも出ていたりして役に立つだろう。ただ，こういった入門書は，機種に依存した内容についてはまったく触れていないので，当たり前だが，マシン語で絵を書いたりキャラクターを動かしたりは無理である。

2冊目は，機種に依存した解析本である。これには，どこの何をどうすると画面に点を打ってくれるとか，どこに何をセットすればROM内のルーチンを使って何をしてくれるのかなどが書いてある。MZやX1については，本誌のバックナンバーを捜すといくらでも情報はあるが，何冊もひっくり返して欲しい記事を捜すのは大変である。結局なにか1冊買うはめになろう。『試験に出るX1』とかね。

ともかくアセンブラは，タダで付いてくるBASICに比べ，金と暇と労力の自己投資が必要なのである。もっとも，自分に投資しない奴はロクなものではない。

理由2：アセンブラは単純である

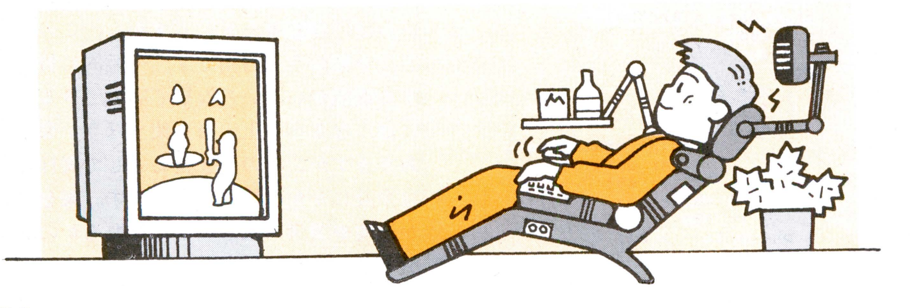
マシン語（正確にはアセンブリ言語）自体は極めて簡単であり単純である。マシン語体操のお兄さんが昔「13語でできるマシン語」なんていったように，基本的なところさえ押さえておけばあとはその組み合わせでたいていのことはできる。

しかしそこが，曲者なのである。とくに，BASICを遅いと感じてマシン語に手を出そうと思う人は用心である。そういう人は複雑なことをやりたいに違いないからだ。

単純なアセンブラで複雑なことをやろうとすると，単純なものを複雑に組み合わせねばならなくなる。つまり，人間側の行為が複雑なのである。逆に，複雑なことが好きな人には，アセンブラは大変な快感となろう。

理由3：アセンブラは覚えにくい

マシン語は覚えにくい。少なくとも，総合的にアセンブラを駆使しようと思うと，自分の記憶力のなささを痛感する。わかりにくいマシン語命令に1対1の名前をつけて人間寄りにしたものがアセンブラなどという解説をまま見かけるが，それはもう半世紀も前の話で，私ら日本人に言わせるとアセンブラなんてアルファベットの記号が並



んでいるだけだ。

Z80でいえば，LDはLoaDの略で，INCがINCrementの略で，CALLはそのものでと，それくらいはわかるけど，LDにも足し算のADDにもいくつも種類があって，初心者がニーモニック表などを眺めた日には「今までの決意はなかったことにしよう」としてパタリと閉じてしまいそうなものだ。わかってくれれば，規則性が見えてくるのでいちいち覚える必要もないのだが，そんなことは初心者の知るところではない。

まあ，命令ひとつにオペランド2つという形が決まっているのがまだしもだろうか。オペランドというのは命令を実行するのに必要な項目のことである。AとBを足すという処理の場合には“足す”が命令で，AとBがオペランドである。アセンブラの世界へ行くと，こういった耳慣れない用語が待っているのを覚悟せねばなるまい。

次に，アセンブラを触ろうと思ったら，まずレジスタのことを知らねばならない。命令は13語だけしか知らなくとも，レジスタは全部知っていないとお話にならない。このレジスタというやつはまた厄介で，AとかBとかいう名前で呼ばれている。たとえば，Z80にはAからFまである，それでもって，HLレジスタ(きっと，HighとLowだろう)やら，IX，IY（インデックスレジスタ）もあり，用途の決まっているSP（スタックポインタ）レジスタやPC（プログラムカウンタ）なんかもあって，てんやわんやである。

なおかつ，2つのレジスタをくっつけて16ビットのレジスタとして使えるなどときた日には，もっとわかりやすいモノは作れなかったのか！　と怒ってしまう。

これが68000になると，D0～D7とA0～A7という名前になって，DがデータでAはアドレスの頭文字かな，とは思うのだけれど，数字で区別されても困ってしまう。

というわけで，アセンブラは覚えにくいというのがわかって頂けたであろうか。

さらに，アセンブラがわかっても，いざプログラムを作ろうと思うと，当然，CPUを取り囲むさまざまな環境を手なずける必要が出てくる。たとえば，字を書くルーチンを呼ぶにはそのアドレスをコールしなければならないのだが，そのアドレスが仮に3F50Hだったとすると，

CALL 3F50H

である。これでは，なんのことか3日もたてば忘れてしまう。しかもその前に字の入っているアドレスをどっかのレジスタ（そのルーチンを作った人が決める）に入れておくなどの準備も必要だ。まあ，アセンブラもよくしたもので，あらかじめ“_PRINT”ときたら“3F50H”だと思えと言い含めることはできる。そうすれば，

CALL _PRINT

でよい。

そして，動機・息切れ・めまい

実をいうと，こんな私とて，何とか楽をしてアセンブラを使いたいと思った日々がなかったわけではない。が，これがまた，シャレにもならない動機，息切れ，めまいの連続であった。

MZ-2500で遊んでいたころ，強力無比なBASICにも，どうしても気に入らないところがあった。すでにあるファイル名でセーブしようとすると，そのファイルはあるからダメだよーん，と拒否されるのである。仕方ないから，KILLしてからSAVEする。これは気に入らない。デリートというのは精神的によくないのだ。そこで，すでにあるファイルと同じ名でセーブするとき専用の命令をつけることにした。BASICの書き換えである。これが動機，だ。

ここで，どうすれば楽ができるかを考える。解析本を眺めると，BASICの命令のテーブルを発見した。同時にそのテーブルを見てその処理をするマシン語のルーチンを呼び出すことがわかったので，いらない命令を新しい命令に変えてしまうことにした。なるべく他に影響を与えなさそうな命令ということで選ばれたのがHCOPYである。私はプリンタを持っていなかったので，不要な命令だったのだ。

まず，HCOPYをRESAVに書き換える。RESAVコマンドプロジェクトの作動である。テーブルのHCOPYをRESAVにすると，次はHCOPYルーチンの書き換えである。HCOPYルーチンの始まるアドレスは解析本に書いてあるので，問題はない。そこに，自分で作ったRESAVルーチンを組み込むのである。

さて，RESAVルーチンの制作である。ここは，楽をして実を取れの基本精神にのっとり，既存のKILLルーチンとSAVEルーチンを続けて呼ぶことにした（怠慢だなあ）。そして，それをアセンブリ言語で書いて，アセンブラなんて持っていなかったから表を見ながら手でマシン語のリストにして，BASICからPOKE文で書き込んで終わり。

なんともいい加減なプログラムだったが，問題なく動作していたのでよかったのだろう。で，そこでマシン語を触るのはやめた。非常に面白かったのだけれども，アセンブラを持っていなかったのだ。ハンドアセンブルなんてもうしたくない。これが息切れ，である。

仮にアセンブラがあったとしても，あの強力なBASICとおいしいアルゴ機能を捨ててCP/Mなどに走ったであろうか？　アルゴ機能にアセンブラがあったら，と思わざるを得ない。これがめまい，である。

アセンブラと仲良くなる条件

アセンブラに立ち向かうにはいささか適性が必要であることに，私はとっくに気づいている。私にその適性がないこともだ。

アセンブラに近寄るためには，もの覚えがよくて，まめにメモを残しておくことができてることが大切だ。私はもの覚えが悪くて，メモを残せない（自分で書いたメモが3日で読めなくなり，1週間でメモの置き場もわからなくなる）人間なのだ。

そして，凝り性でなければならない。アセンブラというのはある程度のめり込まないと面白さが見えてこないのだ。不運なことに私はまったく凝り性ではない。肩でさえ凝らない。

続いて，短気であってはならない。アセンブラで開発をするには苦難が待ち受けている。ああ，昔，MZ-2000のテープでプログラムを作るはめになったときだった。アセンブラをロードしては煙草を吸い，テープを入れ替えてプログラムをロードしてはコーヒーを飲み，アセンブルし，セーブし，リンカをロードしてはビールを飲み，リンクしては煙草を吸い，デバッガをロードしてはまた煙草を吸い，動かないといって，リストをプリントアウトしながらチョコレートを食べ，リストを眺めながらまた煙草を吸い，しまいには「おまえが悪い」などとMZ-2000を殴ったはいいが，ああ，あなたは強かった。痛いのは私の手であって，MZ-2000の筐体は丈夫だった。そのときは，もうアセンブラなんてやるまいと，お天道様に誓ったほどだ。

ついでにいうと，細かいことにこだわらないほうがよい。3つや4つくらい通らないルーチンがあったっていいじゃないか，おかしな動作をしたって可愛いではないか，メモリを無駄に使ったっていいじゃないか。どうせ，人に売るソフトではなし。自分で楽しむだけならたまに暴走しても「アセンブラだから，ま，いいか」くらいでなければならない。余談だけど，ある情報工学科の教授がスーパーマリオ2を見て，「どうし

てこんなに大きなプログラムなのに（ゲームを妨げるような）バグがないんだ！」と驚いたという話がある。そのくらいバグはつきものなのだ。

うん，やっと話がまとまってきたな。では，締めよう。

この世のしくみ

コンピュータ界はメーカー，システム開発者，アプリケーション開発者，ユーザーの4つに分かれている。ユーザーが楽をしようとすると，アプリケーション開発者が苦労をする。どっちも楽をしようとすると，システム開発者が苦労をする。みんなが楽をしようとすると，ハードメーカーが苦労をする。そんな関係なのだ。

みんな，楽をしていいものを使いたいし，いいものを作りたい。本当は，メーカーやシステムの開発者に頑張ってもらって，アセンブラに頼らなくてもハードを活かせたり，統一された環境で快適にプログラムが開発できたりするのが一番よいのだ。シャープさん，頑張ってHumanやCをバージョンアップしてね。

では，若い諸君も，よりよいプログラマになってください。私はアセンブラは嫌いだが，アセンブラで作られた質のいいプログラムは大好きなのだ。

荻窪圭がX68000のアセンブラで作った 最初で最後(?)のプログラム

X68000の初代バージョンには，福袋と称するディレクトリがあり，そこにはアセンブラ，リンカが入っていた。これはおいしい。使わにゃそんそん。そこで考えた。簡単に作れて，X-BASICにはできないこと。んだば，スーパーインポーズでTVに時刻を表示するプログラムでも作ってみるか。

ん？　X-BASICでも［SHIFT］+［+］キーでスーパーインポーズにすればいいではないかと申すのか。それでは，つまらない。それに，実際に使おうと思ったら，いちいちBASICから立ちあげるより，OSからコマンド一発のほうがよいではないか。それに，将来，TV画面にリモコンウィンドウを開いてマウスでTVを操作するプログラムに拡張する予定があったのだ。

そして，リストのプログラムが出来上がった。ただ時刻を表示してもつまらないから，表示場所を左上，右上，右下，左下の4カ所用意してあり，マウスの左ボタンで選択できるようになっている。本誌に掲載されたX68000のIOCS DATA LISTと68000プログラマーズハンドブックがあったおかげである。

このリストは実に楽をして作るアセンブラプログラムとなっている。まず，IOCSコールの番号を名前に割り当てるというのは，すでに常套の型である。それでもって，定数もこのパターンで割り当てておく。アセンブラさんのおかげである。

このプログラムでは，4つの変数が必要である。X，Y座標と，カウンタ（表示場所に0～4の番号をつけてあるのだ），現在時刻である。現在時刻はメモリ上に持っているが，他の3つはD3～D5の3つのレジスタを占有している。レジスタをたくさん持っていて自由に使える68000ならではの技である。

今となってはCがあるおかげでアセンブラを使う必要はなくなったから，久びさにこのプログラムを見たとき，自分で作ったものでありながら読めない，という事態になって困った。拡張するのはやめて，Cで書き直そう。

リスト1　スーパーインポーズによる時刻表示

```
*
* ｽｰﾊﾟｰｲﾝﾎﾟｰｽﾞ ﾃﾞ ｼﾞｺｸ ｦ ﾋｮｳｼﾞ ｽﾙ｡
*
*(D0-D2:WORK,D3:X,D4:Y,D5:COUNTER)
*
*
*/IOCS
*
TVCTRL          equ     $0C
TGUSEND         equ     $0E
CRTMOD          equ     $10
MS_GETDT        equ     $74
MS_OFFTM        equ     $78
*
TIMEGET         equ     $56
TIMEBIN         equ     $57
TIMEASC         equ     $5B
*
B_CUROF         equ     $1F
B_CURON         equ     $1E
B_PUTMES        equ     $2F
B_COLOR         equ     $22
B_LOCATE        equ     $23
B_PRINT         equ     $21
B_CLR_ST        equ     $2A
*
*/OS
*
_EXIT           equ     $FF00
*
*/VALUE
*
ATTRI           equ     $6
X1              equ     2
X2              equ     22
Y1              equ     1
Y2              equ     13
*
*
                bsr     INIT
MAINROOP:
                moveq   #MS_GETDT,D0
                trap    #15                     *ﾏｳｽ ﾎﾞﾀﾝ
                cmpi.w  #$FFFF,D0
                beq     FIN
                cmpi.w  #$FF00,D0
                beq     TIME_LOC
RE_TIME_LOC:    bsr     TIME_PRT
                bra     MAINROOP
TIME_PRT:                                       *ｼﾞｺｸﾋｮｳｼﾞ ﾙｰﾁﾝ
                moveq   #TIMEGET,D0
                trap    #15                     *ｼﾞｺｸﾖﾐｺﾐ
                move.l  D0,D1
                moveq   #TIMEBIN,D0
                trap    #15                     *ｼﾞｺｸﾍﾝｶﾝ
                move.l  D0,D1
                lea     TIMEDAT,A1
                moveq   #TIMEASC,D0
                trap    #15                     *ｼﾞｺｸｰﾓｼﾞﾚﾂ
                move    D3,D1
                move    D4,D2
                moveq   #B_LOCATE,D0
                trap    #15                     *ﾋｮｳｼﾞ
                lea     TIMEDAT,A1
                moveq   #B_PRINT,D0
                trap    #15
                rts
TIME_LOC:                                       *ｼﾞｺｸ ﾋｮｳｼﾞｲﾁ ﾍﾝｺｳ
                cmpi    #4,D5
                beq     LOC0
                cmpi    #3,D5
                beq     LOC4
                cmpi    #2,D5
                beq     LOC3
                cmpi    #1,D5
                beq     LOC2
                blt     LOC1
RE_LOC:         move    #2,D1
                moveq   #B_CLR_ST,D0
                trap    #15                     *ｶﾞﾒﾝｸﾘｱ
                move    #0,D2
                move    #0,D1
MS_JUDGE:       moveq   #MS_OFFTM,D0            *ﾏｳｽﾉ ﾎﾞﾀﾝｦ ﾊﾅｽﾏﾃﾞ ﾏﾂ
                trap    #15
                cmpi    #0,D0
                dbne    D0,MS_JUDGE
                addi    #1,D5
                cmpi    #5,D5
                beq     RES_LOC
                bra     RE_TIME_LOC
RES_LOC:        move    #0,D5
                bra     RE_TIME_LOC
*
LOC0:           move.w  #0,D1
                moveq   #B_COLOR,D0
                trap    #15
                bra     RE_LOC
LOC1:           move.w  #ATTRI,D1
                moveq   #B_COLOR,D0
                trap    #15
                move    #Y1,D4
                bra     RE_LOC
LOC2:           move    #X2,D3
                bra     RE_LOC
LOC3:           move    #Y2,D4
                bra     RE_LOC
LOC4:           move    #X1,D3
                bra     RE_LOC
*
INIT:                                           *ｲﾆｼｬﾗｲｽﾞ
                moveq   #B_CUROF,D0
                trap    #15
                move.w  #3,D1
                moveq   #CRTMOD,D0
                trap    #15
                moveq   #$1F,D1
                moveq   #TVCTRL,D0
                trap    #15
                move.w  #ATTRI,D1
                moveq   #B_COLOR,D0
                trap    #15
                move    #X1,D3
                move    #Y1,D4
                move    #1,D5
                rts
FIN:                                            *END
                move.w  #ATTRI-3,D1
                moveq   #B_COLOR,D0
                trap    #15
                moveq   #B_CURON,D0
                trap    #15
                moveq   #$0F,D1
                moveq   #TVCTRL,D0
                trap    #15
                moveq   #$08,D1
                moveq   #TVCTRL,D0
                trap    #15
                move.w  #16,D1
                moveq   #CRTMOD,D0
                trap    #15
                dc.w    _EXIT
*
*       DATA AREA
*
TIMEDAT         ds.b    9
                end
```

超入門Z80マシン語活用術

Kamon Masato 華門 真人
Mounai Toshiyuki 毛内 俊行
Nishikawa Zenji 西川 善司

Z80を使ったX1やMZなどの8ビットパソコンでは，マシン語入門にもさまざまなアプローチが考えられます。今回は，BASICでマシン語を扱う方法から，アセンブラを使ったゲームプログラミングの実際，そしてハードウェアへのアクセス法として割り込みの手法に関するものまで，フルコースでお届けしましょう。

「マシン語」と聞くとウッと本能的に拒絶反応を起こしてしまう人がいます。気持ちはわからないでもないのですが，マシン語ってそんなに難しいものでしょうか。

マシン語というのはきわめて単純な言語で，かつ万能な言語です。ですから，一度マシン語をマスターしてしまえば，これほど魅力的な言語はありません。まさしく無敵モードを手に入れたようなものです。命令が単純なだけプログラミングが煩雑だというのは確かですが，その代償としてマシン語最大の魅力である高速性と小回りのよさがあるのですから，マニアでなくともなんとかしたい気持ちでいっぱいでしょう。

それでも，マシン語が依然としてマニアックな言語にとどまっているのも事実です。これには，マシン語を使うための環境が多分に影響しているのではないでしょうか。

マシン語環境の実際

マシン語のプログラムを組むには，アセンブラというツールを使うのが基本的な常識となっています。

アセンブラとはアセンブリ言語で書かれたプログラムを，マシン語のコードに変換するソフトウェアのことです。アセンブリ言語はマシン語の命令と1対1に対応していますから，つまり，アセンブリ言語を使うということは，「マシン語でプログラムを書く」ということになるわけです。

Z80マシンの場合，アセンブラを使うにはなんらかのシステム（S-OSやCP/Mなどのこと）を利用することになります。以前は，CP/Mを使うのが当たり前のようにいわれてきましたが，Oh!Xの読者の間ではS-OSを使うのが主流となっています。

さて，問題はこれからです。まずシステムを立ち上げたら，そのシステムからエディタをロードします。これでようやくプログラムを組めるわけです。プログラムを編集したら今度はアセンブラを起動，うまくアセンブルできたらしめたものですが，エラーが出たら再びエディタに逆戻り。そうしてアセンブルができ，一応マシン語プログラムなるものができたら，モニタに戻っていよいよ実行。ふつうは暴走するでしょう。一度でうまく動くほうが珍しいのです。というわけで，またエディタに戻って……。

このように，マシン語の「正しく動く」プログラムを完成させるまでには長い道程があるわけです。ある程度のプログラムが組めるようになった人にとっては，システム的に見て納得がいくかもしれませんが，初心者にとって僅か数行のプログラムを試してみるのにこの煩わしさはいただけません。マシン語の美味しさを味わう前に挫折してしまい，「いやぁ，マシン語はどうも」ということになってしまうのです。

では，どうすればよいのでしょう。マシン語修得は面倒くさいからムリ？　そんなことはありません。必要なのはまず段階を踏んでいくこと。なにごとも基礎からです。

まずはハンドアセンブルから

マシン語の基本中の基本，それはハンドアセンブルです。なにも，思い込んだら試練の道をといっているわけではありません。これは，マシン語を学ぶうえでのひとつの近道でもあるのです。

アセンブラはマシン語の知識が前提ですから，なにも知らない状態でいきなりアセンブラを使おうとしても，アセンブルするソースプログラムが作れなくてはなにもできません。まして下手に使い方のよくわからないアセンブラを使ってしまうと，エラーは続出するし，何度もエディタに戻って修正を繰り返さなくてはならないでしょう。ごく数行であれば，ちゃんとアセンブルできて正しく動くようなソースプログラムが作れるようになるまでは，命令表を見ながらハンドアセンブルしたほうがかえって早いということもあるのです。

もちろん，ちょっとの期間修練すれば，もはやアセンブラなしにはマシン語のプログラムなんて書けないようになるでしょう。しかし，ハンドアセンブルを経験しておくと，なによりもマシン語の働きを身をもって理解できるようになります。アセンブラだとあまり意識しない相対ジャンプと絶対ジャンプの違いなども学ぶことができるでしょう。また，エラーを出さないようにするために自然と，シンプルなわかりやすいプログラムを作る習慣がつくのも利点です。

BASICとマシン語

さて，数行のプログラムをハンドアセンブルで組んだとしたら，モニタから実行というのもいいのですが，もっと有効な方法もあります。そう，BASICと組み合わせて利用するのです。

速くて万能だが扱いの難しいマシン語と，プログラミングは簡単だがスピードが遅く細かい制御ができないBASICの，それぞれの長所をうまく組み合わせれば，なんでもできるし，取り扱いも簡単，そしてスピードもそこそこ速くすることができるはずです。基本的に，スピードを要求される部分や，細かい制御などを行うところではマシン語を使い，そのほか遅くてもよい部分はBASICを使うとよいでしょう。こうすれば，非常に効率よくプログラムが組めます。

また，BASICで組んだ完成したプログラムがあれば，そのサブルーチンをひとつずつマシン語ルーチンに置き換えていけば，完全にマシン語のプログラムも作れるのです。こうして，やがてマシン語に対する苦手意識がなくなったら，アセンブラでバリバリとマシン語プログラミングの本道を行けるようになるでしょう。それではさっそく次のページから実践に入りましょう。

サブルーチンから始めよう

いきなりオールマシン語のプログラムを作れる人はそうそういません。まずはBASICからマシン語のサブルーチンを使うことから始めてみましょう。基本はマシン語の呼び出し方からです。

Kamon Masato 華門 真人

BASICからマシン語を

マシン語を扱うときはアセンブラが常識だが，アマチュアなのだから開発効率なんか気にしないでハンドアセンブルしてみるのもよいものだ。CPUになりきったつもりで命令表を追っていると，アセンブラを使っていたのでは得られないような，マシン語の本質が見えてくる。

ハンドアセンブルは勉強になるとはいえ，あまり大がかりなプログラムには向いていない。作ってもサブルーチンくらいのものだ。しかし，ほんの小さなサブルーチンでもBASICと組み合わせれば，実に手軽で実用性十分。遅い遅いといわれるBASICプログラムでも要になるサブルーチンをマシン語ルーチンに置き換えれば驚くくらい速くなってくれる。ここでは，もっとも手軽なマシン語の使い方を見ていこう。

つまり，メインはあくまでもBASICで，マシン語を補助として使っていくわけだが，BASICからマシン語を使うにはどうすればよいのであろうか。

まず考えつくのはマシン語モニタであろう。マシン語モニタであれば，ハンドアセンブルしたマシン語プログラムでも簡単に打ち込めるし，実行も容易だ。そのほか，開発に役立つようなコマンドもいろいろある。しかしここでは，あくまでBASICプログラムの中でマシン語を使うということに的を絞ろう。となると考えられるのが，PEEK, POKE, CALL, USRなどのBASICコマンドである。

まあ，なにはともあれ，X1を例にこれらのコマンドの使い方を見ていってみよう。

●PEEK/POKE

この2つのコマンドは誰でも名前だけなら知っているかもしれない。それぐらい有名なコマンドである。働きもきわめて単純で，メモリに1バイト書き込む，メモリから1バイト読み込むだけの話である。

すなわち，

　POKE　&HD000,&H80

とすればメモリのD000H番地に80Hというデータを書き込むし，

　A=PEEK(&HD000)

とすれば変数AにメモリのD000Hから1バイト読み込むことになる。

このコマンドをどうやって使うかだが，主にBASICプログラムからマシン語プログラムへの（POKE），またはマシン語からBASICへの（PEEK）データのやりとりに使うことになるわけだ。

●CALL

これもまた有名な命令だよね。マシン語のコマンドでもCALLというのがあるからね。これも働きは簡単。指定されたアドレスのマシン語ルーチンをサブルーチンコールするだけのこと。たとえば，

　CALL　&HD000

とすれば，メモリのD000H以降のマシン語ルーチンをコールするわけ。逆にマシン語ルーチンからBASICに戻るためには，マシン語ルーチンの最後にC9H（RET）を書いておけばよい。すなわち，BASICでいうRETURNというところだ。

●USR

さて，上に述べた3命令だけでも十分マシン語ルーチンを扱うことはできるが，もうひとつ便利な命令がある。このUSRという命令，基本的にはCALLと同じようなものなのだが，詳しく知らない人も多い。実は使い勝手がよかったりするのだが。

USRは基本的にはCALLと同じであると書いた。すなわちUSRもマシン語のルーチンを呼び出す命令には変わりない。その違いは呼び出す形式にある。すなわち，CALLがサブルーチンとしてマシン語ルーチンを呼び出すのに対し，USRは関数として呼び出すのだ。

関数として呼び出す，すなわち，ある値（引数）を持ってマシン語ルーチンに飛び，またある値（返り値）を持って戻ってくるのである。これがCALLよりも使い勝手のいい理由である。CALLでは呼び出すことはできても直接，値を引き渡すことができない（X1ではマニュアルには出ていないが，コールアドレスの次に括弧でくくられたデータを書くことによって，ひとつだけ値をHLレジスタに引き渡せる。しかし裏技なのであまり使わないほうがよいだろう）。そこで値を引き渡すためにPOKE/PEEKを使うことになる。

USRの使い方

というわけでUSRのほうが使い勝手がいいわけだが，それではどうやって使うのか。これは関数として呼び出すのだから，

　A=USR0(B)

のようにすることになる。いうまでもなく，Bが引数，Aが返り値というわけだ。

あれ，呼び出すルーチンのアドレスはどこにあるのだろうか。これは実はあらかじめ定義しておくのだ。たとえば，

　DEFUSR0=&HD000

としておけば上のコマンドで自動的にD000H以降のルーチンに値Bを持ってコールし，返り値をAに入れて戻ってくることになる。ここらへんはDEFFNとFNの関係と同じだ。

なかには少し疑問を感じている人がいるかもしれない。そう，BASICの変数をどうやってマシン語に引き渡して（またはその逆）いるのだろうか。そこらへんのことも含めてもう一度丁寧に説明しよう。

USRの場合呼び出すアドレスをDEFUSRでいちいち指定しなければならないから面倒じゃないかと思った人がいるかもしれない。しかしそれは誤解である。ただUSRといっても実は10個のUSRがあるのだ。すなわちUSR0から始まってUSR1, USR2, そしてUSR9まで10個のUSR関数がある。

この10個のUSR関数に対して，当然DEFUSRもDEFUSR0からDEFUSR9まで10個あるのだ。だからたいていプログラムの先頭で必要なだけ，

　DEFUSR0=～,:DEFUSR1=～,

と定義してしまう。そしてプログラム中ではUSR0などで必要なルーチンを呼び出す。そう，かえっていちいちルーチンのアドレスを書かなくてすむだけ便利なのだ。いうならばCALLが行番号で呼び出すのに対し，USRは0～9というラベルで呼び出すといった感じなのである。

さて，USRの最大の特徴である引数，返り値だが，はたしてどのように伝えられるのであろうか。実は引数のデータをレジスタに書き込んでおくことによってデータを引き渡しているのだ。

その1　y＝USRn(x)

引数・返り値が数値変数の場合。

Aレジスタ：引数の変数型
02$_H$：整数型
05$_H$：単精度型
08$_H$：倍精度型

HLレジスタ：引数の入っているメインメモリのアドレス

つまり，引数はAレジスタとHLレジスタによって与えられる。HLレジスタで示されるアドレス以降の数バイトに引数の内容が書き込まれているのだ。その数バイトというのはAレジスタで示される。Aレジスタ，すなわち変数型というのは実は引数がメモリ内で何バイトで表されているのかを示しているのである。たとえば単精度型ならば5バイトというわけだ。

結局，数値変数の場合は引数はHLレジスタ以降のAレジスタバイトというわけ。また，マシン語ルーチンの中でこの引数の値を書き換えるとそれが返り値yとなって戻ってくるのだ（書き換えなければxがそのままyに入ってくる）。

文字関数を使う場合

その2　y$＝USRn(x$)

引数・返り値が文字変数の場合。

Aレジスタ：引数の変数型
文字型だから当然03$_H$

HLレジスタ：ストリングディスクリプタ

これがちょっと面倒くさいのだが，ストリングディスクリプタとは相対アドレスのことである。文字変数の中身はメモリ上のあるエリアにまとまって記憶されている。その文字記憶エリアの先頭から何バイト目にあるかを示しているのがHLレジスタ以降の内容である。正確にはHLレジスタの示すアドレス以降3バイト（これはAレジスタで示されている）に文字変数の長さ（1バイト），相対アドレス（2バイト，下位，上位の順）が記憶されている。

なんだか面倒くさいなと思ったあなた。大丈夫，ちゃんと逃げ道はある。それがDEレジスタとBレジスタだ。

DEレジスタ：文字データのアドレス

なんのことはないDEレジスタに文字データのアドレスがちゃんと入っているのだ。

Bレジスタ：文字データの長さ

さらにご丁寧なことに長さはBレジスタに入っている。HLレジスタで示されたアドレスの中身を調べても同じだけどね。

というわけで，文字変数の場合引数はDEレジスタ以降Bレジスタバイトなのである。簡単だよね。また引数をマシン語ルーチンの中で書き換えるとそれが返り値y$として返ってくるというのは数値変数のときと同じである（書き換えなければx$の値が入ってくる）。

ただし注意しなければならないのは，返り値のためにもともとの変数を書き換えるとき（すなわちDEレジスタ以降の内容を直接書き換えるとき）。この場合確かに返り値は書き換わったものが戻ってくるけれども，引数までもが書き換わってしまう（データそのものを書き換えてしまうんだから当たり前だ）。

そこで引数を書き換えないためにはどうするかだが，x$という文字変数を引数として与えたい場合には，

y$＝USR0(x$＋“”)

のように引数部分をちょっと工夫してやればよいのだ（“”はヌルストリング）。

こうするとなんで大丈夫なのだろうか。実は＋“”というものを加えてやることにより新たにx$＋“”（実質x$と同じ）という変数ができる。そしてDEレジスタはその新しい変数のアドレスを指すというわけ。こうすればDEレジスタ以降を書き換えてももともとの引数にはなんの変化もないのだ。

ところで数値変数では同じようにして書き換わってしまう心配はないのだろうか。文字変数のように引数まで書き換わってしまうのでは。しかし実際にはこの心配はない。というのも数値変数では自動的に演算用ワークエリア内に引数のコピーをとり，そのコピーをマシン語ルーチンに引き渡すからである。というわけで書き換えても変わるものはコピーと返り値だけ，引数は保護されているのだ。要するに文字変数で＋“”とするのが自動的に行われているということだ。

USRにはCALLにはない利点がまだある。それはマシン語ルーチン内で発生したエラーをBASICに戻って処理できるという点である。やり方は簡単でAレジスタにエラー番号（たとえばType mismatchならば13）を入れ，IXレジスタで示されるアドレスにジャンプ（JP（IX））すればよい。

このIXレジスタの示しているアドレスは実はエラー処理ルーチンで，Aレジスタで示されるエラーをBASICに戻って発生してくれる。これとON ERROR GOTOなどを使えば，マシン語ルーチン内のエラー処理もラクラクというわけだ。

実際にやってみよう

使い方がわかったところでさっそく実際にプログラムを作ってみよう。最初は小さいくだらないプログラムで十分だ。もちろんハンドアセンブル。リスト1はCALLを

リスト1　CALLを使う

```
1000 'Machine language & BASIC
1010 '        program 1
1020 'CALL & POKE & PEEK
1030 '
1040 LIMIT &HD000
1050 'Machine language
1060 FOR I= 0 TO 1000
1070 READ DT$
1080 IF DT$="*" THEN 1350
1090 POKE &HD000+I,VAL("&H"+DT$)
1100 NEXT
1110 DATA 2A,30,D0       :'LD     HL,(D030H)
1120 DATA 54             :'LD     D,H
1130 DATA 5D             :'LD     E,L
1140 DATA 3A,32,D0       :'LD     A,(D032H)
1150 DATA 3D             :'DEC    A
1160 DATA 83             :'ADD    A,E
1170 DATA 5F             :'LD     E,A
1180 DATA 30,01          :'JR     NC,03H
1190 DATA 14             :'INC    D
1200 DATA 46             :'LD     B,(HL)
1210 DATA 1A             :'LD     A,(DE)
1220 DATA 77             :'LD     (HL),A
1230 DATA 78             :'LD     A,B
1240 DATA 12             :'LD     (DE),A
1250 DATA 23             :'INC    HL
1260 DATA 1B             :'DEC    DE
1270 DATA B7             :'OR     A
1280 DATA E5             :'PUSH   HL
1290 DATA ED,52          :'SBC    HL,DE
1300 DATA E1             :'POP    HL
1310 DATA DA,0E,D0       :'JP     C,1200
1320 DATA C9             :'RET
1330 DATA *
1340 'BASIC
1350 INPUT "文字列 =",A$: B$=A$
1360 SD=VARPTR(B$)+1
1370 HL=PEEK(SD)+PEEK(SD+1)*256+STRPTR
1380 MEM$(&HD030,2)=MKI$(HL)
1390 POKE &HD032,LEN(B$)
1400 CALL &HD000
1410 PRINT "変換後 =";B$
1420 END
```

リスト2　USRを使う

```
1000 'Machine language & BASIC
1010 '        program 2
1020 'USR
1030 '
1040 LIMIT &HD000
1050 'Machine language
1060 FOR I= 0 TO 1000
1070 READ DT$
1080 IF DT$="*" THEN 1340
1090 POKE &HD000+I,VAL("&H"+DT$)
1100 NEXT
1110 DATA 6B             :'LD     L,E
1120 DATA 62             :'LD     H,D
1130 DATA 78             :'LD     A,B
1140 DATA 3D             :'DEC    A
1150 DATA 83             :'ADD    A,E
1160 DATA 5F             :'LD     E,A
1170 DATA 30,01          :'JR     NC,03H
1180 DATA 14             :'INC    D
1190 DATA 46             :'LD     B,(HL)
1200 DATA 1A             :'LD     A,(DE)
1210 DATA 77             :'LD     (HL),A
1220 DATA 78             :'LD     A,B
1230 DATA 12             :'LD     (DE),A
1240 DATA 23             :'INC    HL
1250 DATA 1B             :'DEC    DE
1260 DATA B7             :'OR     A
1270 DATA E5             :'PUSH   HL
1280 DATA ED,52          :'SBC    HL,DE
1290 DATA E1             :'POP    HL
1300 DATA DA,09,D0       :'JP     C,1190
1310 DATA C9             :'RET
1320 DATA *
1330 'BASIC
1340 DEFUSR0=&HD000
1350 INPUT "文字列 =",A$
1360 B$=USR0(A$+"")
1370 PRINT "変換後 =";B$
1380 END
```

使ったもの。リスト2はUSRを使ったものである。どちらもやっていることは簡単で，単に与えられた文字変数の文字の順を引っくり返しているだけである。はっきりいってなんにも役に立たないプログラムだが最初はこれでいいのだ。完成すれば役に立つプログラムでも結局完成しないんじゃ意味はない。それよりもくだらないプログラムでも動いたほうが遙かに勉強になるはずだ。

というわけでリスト1と2をよく研究してみてほしい，CALLとUSRの違いに気をつけるように。なお，2バイトのデータをPOKEで書き込む代わりにMEM$(〜)＝MKI$(〜)を使っている（こっちのほうが洗練されている）が，ここらへんは自由課題。自分で研究してみよう。

その先にあるもの

さて，ハンドアセンブルで何本もプログラムを作ったら，もうマシン語は怖いものではなくなっているはずだ。マシン語がどういうものかはだいぶわかってきているはずだし，コマンドもだいたい理解できていると思う。

こうなってしまえばしめたものだ。このレベルに達していればアセンブラを理解するのも容易だ。当然アセンブラもラクラク扱えるようになっていると思う。ここまできて初めてアセンブラを使うことに意味が出てくるのだ。

ハンドアセンブルでは作れるプログラムの大きさに限界がある。本格的なプログラムになればなるほどアセンブラの必要性が出てくるのだ。しかしアセンブラを使うようになってもハンドアセンブルで鍛えている人間は強い。コンパクトに，しかもわかりやすくプログラムを作ることを知っているからだ。

こうしていけばマシン語プログラムなんてどんどん作れるようになっていくはずだ。そのためにはBASICがいつでも役立つ。なにも見栄をはってオールマシン語にトライする必要はないのだ。アセンブラで組んだマシン語プログラムをBASICで利用してもよい。要はいきなりオールマシン語ではなく，サブルーチン1つひとつを少しずつマシン語化していくことだ。

リスト3がそれである。これはちょっとしたゲームである（残念ながらハード上の都合でX1turboでしかできないが）。どんなゲームかというとテンキーで指の体操をやってもらおうというのだ。マシン語部は同時キー入力（これはマシン語でないとできない）をやっていて，BASICは残りの部分を担当している（同時キー入力を行っているのでキーボードをBモードにしておくように）。昔，これを体でやるゲームもあったと思う。

やり方は簡単，ディスプレイで指示される通りにテンキーを同時に押していけばよいのだ。最後には5本の指がテンキー上でからみあうことになる。なお，このプログラム，実はハードの構造上けっこういいかげんな処理を行っているが，そこらへんは目をつむっていただきたい。

このようにして，徐々に必要に応じて進んでいけばよいのだ。そしていずれ，すべてをマシン語で記述する必要が出てきたときにオールマシン語に挑戦すればよいのだ。そのときはきっとそれだけの力がついているはずだから。

リスト3　マシン語サブルーチンの例

```
1000 'Machine language & BASIC
1010 '        program 3
1020 '
1030 LIMIT &HD000: LOADM "ML & B.Bin"
1040 DEFUSR0=&HD000: DEFINT A-Z
1050 FG$(1)="親指": FG$(2)="人さし指"
1060 FG$(3)="中指": FG$(4)="薬指"
1070 FG$(5)="小指"
1080 P(1)=&B100000
1090 P(2)=&B1000
1100 P(3)=&B1
1110 P(4)=&B1000000
1120 P(6)=&B10
1130 P(7)=&B10000000
1140 P(8)=&B10000
1150 P(9)=&B100
1160 FOR I=1 TO 5
1170 SF=INT(RND*9)+1
1180 FL=0: FOR J=1 TO I
1190 IF (SF=SF(J)) OR (SF=5) THEN FL=1
1200 NEXT
1210 IF FL THEN 1170 ELSE SF(I)=SF
1220 NEXT
1230 PRINT "指示に従ってテンキーを押して下さい。"
1240 PRINT "  一度指示があったら二度と離さないこと。": PRINT ""
1245 CALL &HD036
1250 FOR I=1 TO 5
1260 PRINT "指示";I;":";FG$(I);"で";SF(I);"キーを押して下さい。"
1270 PRINT "Time limit = 5 sec";CHR$(&H1D,&H1D,&H1D);
1280 FOR J=4 TO 0 STEP -1
1290 PAUSE 10
1300 PRINT CHR$(&H1D,&H1D,&H1D);J;
1310 NEXT
1320 PRINT "": FP=FP OR P(SF(I))
1330 TF=USR0(FP)
1340 IF (TF AND FP) XOR FP THEN 1370
1350 NEXT
1360 BEEP : PRINT "ＯＫ！よくできました。": CALL &HD03F: END
1370 BEEP : PRINT "ブーッ！失格です。": CALL &HD03F: END
```

```
D000 3E E3 CD 12 D0 CD 25 D0 : 92
D008 CD 25 D0 F5 CD 25 D0 F1 : 6A
D010 77 C9 F3 F5 01 01 1A ED : 31
D018 78 E6 40 20 FA 01 00 19 : D2
D020 F1 ED 79 FB C9 F3 01 01 : 10
D028 1A ED 78 E6 20 20 FA 01 : A0
D030 00 19 ED 78 FB C9 3E E4 : 64
D038 CD 12 D0 AF C3 12 D0 3E : 41
D040 E4 CD 12 D0 3E 1A C3 12 : C0
D048 D0                      : D0
--------------------------------
SUM: 86 89 90 F4 7D FC DB FD 22CF
```

```
0000               ;Machine language & BASIC
0000               ;
D000                         ORG     0D000H
D000
D000               KEYIN
D000 3E E3                   LD      A,0E3H
D002 CD 12 D0                CALL    TRANS49
D005 CD 25 D0                CALL    RECV49
D008 CD 25 D0                CALL    RECV49
D00B F5                      PUSH    AF
D00C CD 25 D0                CALL    RECV49
D00F F1                      POP     AF
D010 77                      LD      (HL),A
D011 C9                      RET
D012               TRANS49
D012 F3                      DI
D013 F5                      PUSH    AF
D014 01 01 1A                LD      BC,01A01H
D017               IBFCK
D017 ED 78                   IN      A,(C)
D019 E6 40                   AND     40H
D01B 20 FA                   JR      NZ,IBFCK
D01D 01 00 19                LD      BC,01900H
D020 F1                      POP     AF
D021 ED 79                   OUT     (C),A
D023 FB                      EI
D024 C9                      RET
D025               RECV49
D025 F3                      DI
D026 01 01 1A                LD      BC,01A01H
D029               OBFCK
D029 ED 78                   IN      A,(C)
D02B E6 20                   AND     20H
D02D 20 FA                   JR      NZ,OBFCK
D02F 01 00 19                LD      BC,01900H
D032 ED 78                   IN      A,(C)
D034 FB                      EI
D035 C9                      RET
D036               NOINT
D036 3E E4                   LD      A,0E4H
D038 CD 12 D0                CALL    TRANS49
D03B AF                      XOR     A
D03C C3 12 D0                JP      TRANS49
D03F               USINT
D03F 3E E4                   LD      A,0E4H
D041 CD 12 D0                CALL    TRANS49
D044 3E 1A                   LD      A,01AH
D046 C3 12 D0                JP      TRANS49
D049
```

ゲームはやっぱりアセンブラ

まとまったプログラムをマシン語で組むにはやはりアセンブラが必要になります。ここでは，アセンブラを使ってマシン語ゲームを作るためのノウハウを，実際のプログラムを見ながら説明していきましょう。

Mounai Toshiyuki 毛内 俊行

今月はマシン語特集とリンクして，S-OS “SWORD”上で動く全機種共通の新型エディタアセンブラが発表になりました。そこで，このコーナーではアセンブラによるプログラミングの基礎を学ぼうというわけで，簡単なゲームプログラムを作ってみたいと思います。

ここではZ80用の一般的なアセンブラにそって解説しますが，掲載するプログラムは全機種共通にするため，実行にはS-OS “SWORD”が必要です。まだ“SWORD”をお持ちでない方は117ページのX1版“SWORD”のコーナーをご覧ください。

なぜアセンブラなのか

マシン語を使うことの魅力はなんといってもスピードが速いということです。X1やMZなどの8ビット機でスピードの要求されるプログラムを開発するには，アセンブラを使うのが常識でしょう。しかし，実行速度だけを考えればコンパイラを使えばいいようにも思えます。ではなぜ，記述の便利な高級言語によるコンパイラが，Z80などの8ビットマシンでは主流にならなかったのでしょう。その答えのひとつはこれから紹介するプログラムリストのなかにあります。そう，アセンブラで開発するマシン語プログラムでは，プログラムサイズがとても小さいのです。ほかの言語ではこのように小さくはなりません。これはメモリの制約の大きい8ビットマシンでは重要なことで，まさにこれこそアセンブラの魔力といえるのではないでしょうか。

これだけは覚えよう

まずは，Z80のアセンブラで書かれたプログラムを理解するために，最低限知っておきたい疑似命令を整理しておきましょう。疑似命令は，マシン語そのものの命令ではなく，アセンブル時に宣言されるだけの命令，はやい話が，アセンブラというソフトに対する命令というわけです。疑似命令は，プログラムの中で直接実行されるものではありませんが，プログラムを作るうえで最も大切な命令ですので覚えておいてください。

ORG命令

アセンブラが出力するマシン語プログラムを，メモリ上に展開するときの先頭アドレスを定義します。たとえば，

　　ORG　8000H

と書かれていたら，それ以降のプログラムは8000H からメモリ上に生成されます。指定がないとシステムが破壊されます。

ラベル

これは，BASICなどで使っているラベルと同様のものです。たとえば，

　　MAIN:

と書かれていたら，ここがMAINというラベルのエントリ(入り口)になるのです。ラベルのあとの「：」は，ここでラベルが終了することを宣言するものですが，省略してもかまいません。

EQU命令

プログラムの中で，アドレス定義がされていないラベルの定義を行う命令です。これは一種の代入文のような命令で，たとえば，

　　#LOC:　EQU　201EH

とすると，#LOCというラベルに，201EH という値が定義されます。

データ

メモリ上には，プログラムだけでなく，実行に必要なデータを確保しなければなりません。アセンブラにはそのための命令がいくつか用意されています。その代表的なものが，DEFB命令です。たとえば，

　　DEFB　30H

とすると，メモリ上に30H というデータをセットします。これは1バイト単位でのデータのセットですが，このほかに2バイト単位でデータをセットするDEFW命令や，文字列データを扱うDEFM命令があります。また，特定のデータはセットしませんが，とりあえずデータの領域を確保するDEFS命令もあります。

高級言語のテクニックを盗め!

BASICやCなどの高級言語も，もとをただせばマシン語が動作させているものです。つまり，高級言語で使えるテクニックがマシン語で使えないわけはありません。ちょっと代表的なテクニックを紹介しておきましょう。

変数

アセンブラには変数という概念がありません。変数とよく似た使い方をするものとしてレジスタが存在しますが，レジスタは

S-OS通信　その1

S-OS “SWORD” といえば，泣く子も黙る本誌読者の合言葉，じゃなくて全機種共通プログラムの中核となるシステムの名前ですが，本誌の名前がOh!Xになって以降，新しく読者になられた方にはなんのことかさっぱりわからないかもしれません。

簡単に説明しましょう。S-OSというのは，MZ やX1 の各機種に備わっているモニタのサブルーチンコールのエントリアドレスを共通化するサブルーチン集です。えっ，わからないって？　つまりですね，それぞれのパソコンには，ハードウェアの機能を制御する基本ルーチンが用意されているわけです（BIOSとかIOCSとか呼ばれるのがそう）。これらのルーチンはキー入力や画面表示とかのためのもので，どのマシンでも同じような機能のルーチンが並んでいるのです。ところが，これらのルーチンの呼び出し方が機種ごとに違ったりするのです。同じBEEP音を持つパソコンであっても，音を鳴らすには機種ごとに違うアドレスをコールしなければならず，こういうことが積み重なってぜんぜん違うプログラムになってしまうのです。悲しいですね。

そこで，本誌では，それらのサブルーチンコールの仕方を統一するプログラムを各機種に用意しました。それがS-OSなのです。こうしてS-OSが載った機種ではマシン語プログラムも共通です。たとえばBEEP音を鳴らしたければ，プログラムは各機種のS-OSに向かって1FC4H をコールします。するとS-OSは各機種ごとに決められた呼び出し方にしたがって「BEEP音をくれないか」と伝えてくれるわけです。

こうして，今月号で発表となったエディタアセンブラもS-OSの動くすべての機種で共通となっているわけです。そういうわけですから，まだS-OSをお持ちでない皆さんはぜひとも入力して全機種共通システムの世界に参加してください。

（S-OS宣伝担当）

それほどたくさんあるわけではなく，また
プログラムの実行中に，レジスタの内容を保存しておくのは大変です。

そこで，アセンブラでプログラムを書く場合には，データをセットしておくためのワークエリアを，あらかじめメモリ上に確保しておきます。これは，CやFORTRANなどでいう変数の宣言だと思っていいでしょう。たとえばリスト中に，

```
TIME:  DEFS 1
```

としておけば，TIMEという名の変数(実際にはラベルですが）を宣言したことになります。この変数を使いたいときには，

```
LD   A，(TIME)
```

として，Aレジスタに変数を呼び出せばいいのです。変数を書き換えたいときは，変数の処理が終了した時点で，再びワークエリアに戻しておきましょう。

短いプログラムを作るときは，なにかとレジスタだけで実行しようと考えがちですが，ワークエリアを有効に使ったほうがプログラムが見やすくなり，モジュール化も容易になります。ワークエリアをばかにしてはいけません。

条件

高級言語には条件判断命令がいくつもあり，その用途によって使い分けられていますが，条件判断の基本はやはり，IF～THEN～です。マシン語では，条件分岐はすべてフラグレジスタによって行われますから，BASICなどのように変数による分岐を行うときは，あらかじめ変数の内容を比較しておかなくてはなりません。たとえば，

```
IF  POS=0  THEN  "LOOP"
```

というプログラムがあったとき，これをアセンブラで書くと次のようになります。

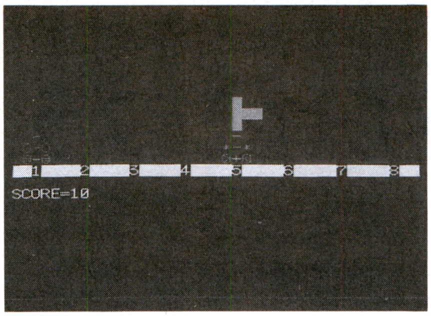
```
LD      A，(POS)
CP      0
JP      Z，LOOP
```

つまり，とりあえず変数をAレジスタに代入して，CP命令で比較すればいいのです。ここで，CP命令を複数使用すれば，CASE文のような分岐も可能であることがおわかりでしょう。

ループ

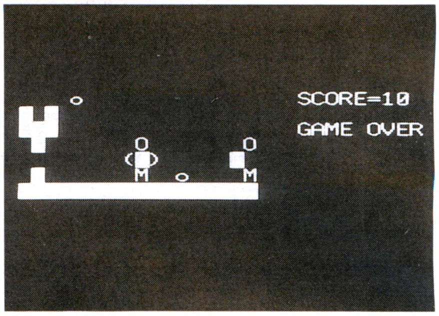
条件分岐とともに必要な命令がループです。これにはいくつかの型があります。まずは，FOR～NEXTのようにループ回数が一定のもので，次のように使われます。

```
        LD      B，10
LOOP:
          (実行部)
        DEC     B
        JR      NZ，LOOP
```



もぐらたたき

【遊び方】
モグラ君が出てきた穴の番号のキー([1]～[8])を押してください。モグラ君の出てくる回数は50回。君は何発当てることができるか。



ボールマン

【遊び方】
[4]，[6]（[Z]，[C]または[←]，[→]でもよい）キーでボールマンを左右に動かし，コーチの出すボールをトスしてバスケットゴールに運び込もう。

これは，実行部を10回実行するプログラムですが，「DEC　B」と「JR　NZ」は，DJNZ命令にまとめられるので，

```
        LD      B，10
LOOP:
          (実行部)
        DJNZ    LOOP
```

とすることもできます。

それから，REPEAT～UNTILのようなものも覚えておくとよいでしょう。プログラムで表すと，

```
LOOP:
          (実行部)
        LD      A，(TIME)
        CP      0
        JR      NZ，LOOP
```

というような形になります。この例では，TIMEというワークの中身が0にならない限り，実行部を繰り返すというものです。

このほかに条件判断を実行部の前で行う，WHILE～WEND型のループがありますが，基本はREPEAT～UNTILと同じなので省略します。それぞれ用途に応じた使い分けをしてください。

いよいよゲームだ!

さて，これまで説明してきた基本テクニックが理解できたら，それらが実際のプログラムの中でどのように使われているかを見ていきましょう。マシン語のプログラムを作るとなると「やはりゲームを」という声が聞こえてきそうです。

では，どんなゲームを作ったらいいでしょうか。まさかいきなり「スペハリ」を作ろうなんて言わないですよね。まずは小さいプログラムから始めましょう。誰だってそうなんです。私だってアセンブラを使い始めたころは，50バイトにも満たないプログラムを作っては「うーん，名作ができた」とほくそ笑んでいたものです（BASICを覚えたころもそうだったでしょ？）。

ただ，Oh!Xに載せるマシン語入門のサンプルですから，小さくて単純なプログラムでもそれなりに面白いゲームを作らなくてはなりません。すっかり悩んでしまった私を救ってくれたのが「ゲームウォッチなんかいいんじゃない」という編集のUさんの一言でした。その言葉を聞いた瞬間，私の記憶の彼方からいくつものゲームがピコピコと音を立てて蘇ってきたのです。

ゲームウォッチといえば，まだファミリーコンピュータが登場する前に一世を風靡したものですが，僅かのキャラクタパターンを使ったごく単純なゲームながらずいぶんと熱中させられたものです。そこで，パソコンの画面をゲームウォッチの液晶画面に見立てて，単純な動作で遊べるゲームを考えてみました。こうして，生まれたのが今回作る「もぐらたたき」と「ボールマン」というわけです。

もぐらたたき

それでは最初に「もぐらたたき」のプログラムを考えましょう。このゲームの舞台は8つの穴のあいた地面で，登場キャラクタは主人公のモグラ君と，そのモグラを叩くハンマーです。モグラ君は，8つの穴のうちのどれかひとつから頭を出してきます。モグラ君が頭を出している間に，頭をハンマーで叩くとポイントになります。モグラ君が50回頭を出すとゲームオーバーになるようにします。

これだけ説明すると，もうこのゲームのプログラムがすべて見えてきます。まず，1～8までの番号をつけた穴を用意して，どの穴にモグラを表示するかを乱数によっ

て決定します。モグラを表示したら１～８までのキー入力を待ちます。もし，押されたキーの番号とモグラの出ている穴の番号が同じなら，ポイントに１を加算して次のモグラの出てくる穴を決定します。これを50回繰り返せばいいわけです。どうです，簡単でしょ？

ボールマン

さて，もうひとつのゲームは「ボールマン」です。このゲームの舞台はどこかの体育館（きっと）で，登場人物は，主人公のプレイヤーとトレーニングコーチの２人です。内容はバレーボールに似ていて，コーチが出すボールの落下点に素早く移動しトスするという単純なもので，３回トスに成功するとボールはめでたくバスケットゴールに入るというわけです。

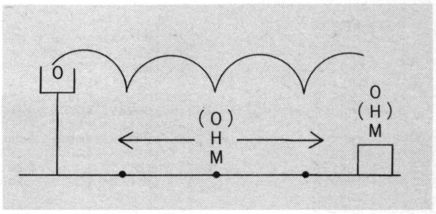
初めのうちは，ボールが１個だけなので右から左に向かって順番に移動すれば，簡単にボールをゴールまで運べますが，根が意地悪なコーチはだんだんボールの数を増やしていき，最後にはボールのスピードまで速くなってしまうというものです。

このプログラムは一見難しそうですが，基本は先ほどの「もぐらたたき」と同じです。まず，ボールの通る道筋をあらかじめ決めておき，その途中に３カ所のチェックポイントを設けておきます。プレイヤーはこのチェックポイント上にあるボールをトスできるようにしておけばいいのです。これを前の「もぐらたたき」と比べると，ボールのチェックポイントが穴，ボールがモグラ，そしてプレイヤーがハンマーになると思いませんか？

プログラムに挑戦

とまあ，こんな具合です。「ナンセンスなゲームだ」といってしまえばそれまでですが，昔のゲームウォッチにだって，「ヘルメット」だとか「マンホール」とかいったかなりナンセンスなゲームがありましたよね。でもゲームの面白さの秘密っていうのはこういう単純なゲームのなかに隠されているのだと思います。

図１　ボールマンとボールの動き



完成したプログラムはどちらも１Kバイトにも満たないコンパクトなものです。S-OS“SWORD”をお持ちの方ならダンプリストを打ち込めばすぐにも遊ぶことができますが，その前にソースリストをゆっくりと見てください。特に「もぐらたたき」のソースリストにはプログラムの解説を設けました。「ボールマン」でも基本的なテクニックはほとんど同じなので「もぐらたたき」と照らし合わせて理解してください。

今回はせっかくのマシン語入門ですから，遊ぶだけでなく，好きなようにソースを改造したり，オリジナルゲームを作ってみるとよいでしょう。それにはきっと今回の新しいアセンブラが役に立つでしょう。

プログラムの解説

もぐらたたき

A）アセンブラでプログラムを書くときの初期宣言。S-OSの内部ルーチンとコントロールコードのラベル定義，そして8000_Hからプログラムを書くことを宣言している。

B）プログラムを実行するときの初期設定。#WIDCHは画面の桁数を決定するS-OSの内部ルーチン，INITはワークエリアの初期化ルーチンで，これを実行することにより，以降のプログラムで誤動作するのを防いでいる。

C）ここは，条件分岐とループ命令によって構成されている。GETKYはリアルタイムキー入力ルーチンで，Aレジスタに入力されたデータを返してくる。この後，SHIFT＋BREAKが押されたらS-OSのホットスタートへジャンプするようになっている。これは，マシン語プログラムは実行中に，強制的に処理を中断することができないため。

SHIFT＋BREAKのチェックの次は，スペースキーの入力チェック。スペースキーが押されていなければ，もう一度ループを実行する。

D）ゲームのメインプログラム部。MOGPRTはモグラの発生，HAMMERはハンマーの処理，MOGRESはモグラが穴にもぐるルーチンで，メインプログラムは以上３つのサブルーチンを順番に呼び出すだけ。このように，各処理ごとにプログラムをサブルーチン化すると，プログラミングやデバッグが楽になる。

また，「LD　HL，TIME」以降は，変数による条件判断が行われ，REPEAT～UNTIL型のループ構造になっている。この部分をBASICで表現すると，

```
10  REPEAT
20  GOSUB "MOGPRT"
30  GOSUB "HAMMER"
40  GOSUB "MOGRES"
50  TIME=TIME-1
60  UNTIL TIME=0
```

というようになる。TIMEの初期値は初期化ルーチンで50に定義されているので，このループは50回実行されることになる。

E）リアルタイムキー入力ルーチン。ここでは事実上S-OSの#GETKYを呼んでいるだけだが，もうひとつ隠れた動作がある。「CALL RAND16」というのがそれで，RAND16は乱数発生ルーチンだが，乱数がパターン化しないようまったく関係のないサブルーチンから呼び出しだけを行っている。

F）モグラを発生するサブルーチン。乱数によりモグラの出る穴を決定しているが，RAND16が０～15までの乱数を発生させるのに対し穴の数が８個しかないので，「SRL A」を実行して乱数を２で割っている。

実際にモグラを表示するにはXLOCで画面のX座標を求めている。MPRTは，DEレジスタペアの示すアドレスから17バイト分のデータを画面に表示するサブルーチンで，DEレジスタにセットするアドレスを連続的に変えることによってモグラが穴から出てくる感じを表現している。これらの手法はMOGRESルーチンでも用いられ，同様にモグラが穴にもぐるのを表現している。

G）ハンマーの処理を行うサブルーチン。このルーチンはループ構造になっていて，ループ回数はWTIMEに代入され，モグラが地上に顔を出す時間を決定している。

ループの内部では，キー入力がされないときはHA2へジャンプし，表示されているハンマーを消去してHA3へ。キーが押されていてもモグラがハンマーに当たっていないときは，直接HA3にジャンプして，もう一度ループを実行するかの判断をする。

モグラにハンマーが命中すると，SCOREの値に１を加算するが，ここではSCOREの計算をBCD演算という特殊な方法で行っている。リストを見ると，「INC A」，「DAA」という命令がある。このDAAというのがくせもので，たとえば，Aレジスタの内容が09_Hだったときに，まず「INC A」を実行すると，Aレジスタの内容は$0A_H$になる。ところが，その直後にDAAを実行するとAレジスタに06_Hが加算され，内容が10_Hに化けてしまい，あたかも10進数演算を行っているように見えるわけだ。

このBCD演算はメモリ効率や実行速度の面で問題があり，大きなプログラムではあまり使われていないが，ちょっとしたプログラムではこのように手軽に使えて便利なので覚えておくとよい。

H）Aレジスタの０～15の乱数をセットするサブルーチン。乱数の発生には乱数表やRレジスタを使う方法などが簡単でよく使われるが，ここでは計算によって乱数の発生を行っている。基本的には乱数用のワークに入っているデータをビット単位で適当に操作し，その下位４ビットを出力しているだけ。

ボールマン

基本的なテクニックは「もぐらたたき」と同様。ひとつだけ大きく異なるのはボールを表示するときの画面のXY座標の決定の仕方だ。「もぐらたたき」では，XLOCルーチンの中でCASE文のような複数の条件分岐を行っていた。一方この「ボールマン」では，BLOCというルーチンで同様の動作を行っているが，このルーチンではBXYというアドレスから，あらかじめセットしてある座標を読んでくるという方法をとっている。このようにすると，プログラムが短くなり，また実行時間も一定となる。

そのほか，Aレジスタに０を代入するのに，「LD A，０」を使わず「XOR A」を使っている。もちろんこれはAレジスタの場合のみ有効だが，使い方次第ではもっと面白いこともできるだろう。

リスト1　もぐらたたき（ソースプログラム）

```
0000                    ;
0000                    ;ﾓｸﾞﾗﾀﾀｷ ｹﾞｰﾑ    by T.Mounai
0000                    ;
1FFA P                  #HOT:    EQU     1FFAH
1FF4 P                  #PRINT:  EQU     1FF4H
1FE5 P                  #MSX:    EQU     1FE5H
1FD0 P                  #GETKY:  EQU     1FD0H
1FC4 P                  #BELL:   EQU     1FC4H
1FC1 P                  #PRTHX:  EQU     1FC1H
1FB8 P                  #HEX:    EQU     1FB8H
201E P                  #LOC:    EQU     201EH
2030 P                  #WIDCH:  EQU     2030H
0000                    ;
000C P                  CLS:     EQU     0CH
001B P                  BRK:     EQU     1BH
001D P                  LFT:     EQU     1DH
001F P                  DWN:     EQU     1FH
0000                    ;
8000                             ORG     8000H
8000                    ;
8000 3E 28                       LD      A,40
8002 CD 30 20                    CALL    #WIDCH
8005 CD 47 80           MAIN:    CALL    INIT
8008 21 00 11                    LD      HL,1100H
800B CD 1E 20                    CALL    #LOC
800E 11 5C 82                    LD      DE,STDAT
8011 CD E5 1F                    CALL    #MSX
8014                    ;
8014 CD 7D 80           START:   CALL    GETKY
8017 FE 1B                       CP      BRK
8019 CA FA 1F                    JP      Z,#HOT
801C FE 20                       CP      20H
801E 20 F4                       JR      NZ,START
8020 CD 55 80                    CALL    DISP
8023                    ;
8023 CD 84 80           REPEAT:  CALL    MOGPRT
8026 CD 50 81                    CALL    HAMMER
8029 CD AE 80                    CALL    MOGRES
802C 21 6C 82                    LD      HL,TIME
802F 35                          DEC     (HL)
8030 20 F1                       JR      NZ,REPEAT
8032                    ;
8032 21 00 10                    LD      HL,1000H
8035 CD 1E 20                    CALL    #LOC
8038 11 4C 82                    LD      DE,GVDAT
803B CD E5 1F                    CALL    #MSX
803E 06 05                       LD      B,5
8040 CD C4 1F           BREP:    CALL    #BELL
8043 10 FB                       DJNZ    BREP
8045 18 BE                       JR      MAIN
8047                    ;
8047 3E 00              INIT:    LD      A,0
8049 32 6F 82                    LD      (KEY),A
804C 32 6E 82                    LD      (SCORE),A
804F 3E 32                       LD      A,50
8051 32 6C 82                    LD      (TIME),A
8054 C9                          RET
8055                    ;
8055 3E 0C              DISP:    LD      A,CLS
8057 CD F4 1F                    CALL    #PRINT
805A 21 00 0D                    LD      HL,0D00H
805D CD 1E 20                    CALL    #LOC
8060 11 CC 81                    LD      DE,LIN
8063 CD E5 1F                    CALL    #MSX
8066 CD 6A 80                    CALL    SCPRT
8069 C9                          RET
806A                    ;
806A 21 00 0F           SCPRT:   LD      HL,0F00H
806D CD 1E 20                    CALL    #LOC
8070 11 45 82                    LD      DE,SCDAT
8073 CD E5 1F                    CALL    #MSX
8076 3A 6E 82                    LD      A,(SCORE)
8079 CD C1 1F                    CALL    #PRTHX
807C C9                          RET
807D                    ;
807D CD A9 81           GETKY:   CALL    RND16
8080 CD D0 1F                    CALL    #GETKY
8083 C9                          RET
8084                    ;
8084 CD A9 81           MOGPRT:  CALL    RND16
8087 CB 3F                       SRL     A
8089 3C                          INC     A
808A                    ;
808A 32 70 82                    LD      (POS),A
808D CD 1C 81                    CALL    XLOC
8090 26 0A                       LD      H,10
8092 CD 1E 20                    CALL    #LOC
8095 11 FC 81                    LD      DE,MOGDT2
8098 CD D2 80                    CALL    MPRT
809B CD 1E 20                    CALL    #LOC
809E 11 03 82                    LD      DE,MOGDT3
80A1 CD D2 80                    CALL    MPRT
80A4 CD 1E 20                    CALL    #LOC
80A7 11 0A 82                    LD      DE,MOGDT4
80AA CD D2 80                    CALL    MPRT
80AD C9                          RET
80AE                    ;
80AE 3A 70 82           MOGRES:  LD      A,(POS)
80B1 CD 1C 81                    CALL    XLOC
80B4 26 0A                       LD      H,10
80B6 CD 1E 20                    CALL    #LOC
80B9 11 03 82                    LD      DE,MOGDT3
80BC CD D2 80                    CALL    MPRT
80BF CD 1E 20                    CALL    #LOC
80C2 11 FC 81                    LD      DE,MOGDT2
80C5 CD D2 80                    CALL    MPRT
80C8 CD 1E 20                    CALL    #LOC
80CB 11 F5 81                    LD      DE,MOGDT1
80CE CD D2 80                    CALL    MPRT
80D1 C9                          RET
80D2                    ;
80D2 06 11              MPRT:    LD      B,17
80D4 1A                 MP1:     LD      A,(DE)
80D5 13                          INC     DE
80D6 CD F4 1F                    CALL    #PRINT
80D9 10 F9                       DJNZ    MP1
80DB                    ;
80DB 01 50 00                    LD      BC,0050H
80DE 10 FE              MP2:     DJNZ    MP2
80E0 0D                          DEC     C
80E1 20 FB                       JR      NZ,MP2
80E3 C9                          RET
80E4                    ;
80E4 3A 70 82           MOGHIT:  LD      A,(POS)
80E7 CD 1C 81                    CALL    XLOC
80EA 26 0A                       LD      H,10
80EC CD 1E 20                    CALL    #LOC
80EF 11 1B 82                    LD      DE,MOGDT5
80F2 CD E5 1F                    CALL    #MSX
80F5 C9                          RET
80F6                    ;
80F6 3A 6F 82           HSET:    LD      A,(KEY)
80F9 CD 1C 81                    CALL    XLOC
80FC 2C                          INC     L
80FD 26 07                       LD      H,7
80FF CD 1E 20                    CALL    #LOC
8102 11 2D 82                    LD      DE,HDAT1
8105 CD E5 1F                    CALL    #MSX
8108 C9                          RET
8109                    ;
8109 3A 6F 82           HRES:    LD      A,(KEY)
810C CD 1C 81                    CALL    XLOC
810F 2C                          INC     L
8110 26 07                       LD      H,7
8112 CD 1E 20                    CALL    #LOC
8115 11 39 82                    LD      DE,HDAT2
8118 CD E5 1F                    CALL    #MSX
811B C9                          RET
811C                    ;
811C FE 01              XLOC:    CP      1
811E 28 1B                       JR      Z,XL1
8120 FE 02                       CP      2
8122 28 1A                       JR      Z,XL2
8124 FE 03                       CP      3
8126 28 19                       JR      Z,XL3
8128 FE 04                       CP      4
812A 28 18                       JR      Z,XL4
812C FE 05                       CP      5
812E 28 17                       JR      Z,XL5
8130 FE 06                       CP      6
8132 28 16                       JR      Z,XL6
8134 FE 07                       CP      7
8136 28 15                       JR      Z,XL7
8138                    ;
8138 2E 24                       LD      L,36
813A C9                          RET
813B 2E 01              XL1:     LD      L,1
813D C9                          RET
813E 2E 06              XL2:     LD      L,6
8140 C9                          RET
8141 2E 0B              XL3:     LD      L,11
8143 C9                          RET
8144 2E 10              XL4:     LD      L,16
8146 C9                          RET
8147 2E 15              XL5:     LD      L,21
8149 C9                          RET
814A 2E 1A              XL6:     LD      L,26
814C C9                          RET
814D 2E 1F              XL7:     LD      L,31
814F C9                          RET
8150                    ;
8150 3E 40              HAMMER:  LD      A,40H
8152 32 6D 82                    LD      (WTIME),A
8155                    ;
8155 CD 7D 80           HA1:     CALL    GETKY
8158 FE 00                       CP      0
815A 28 43                       JR      Z,HA2
815C CD B8 1F                    CALL    #HEX
815F 38 3E                       JR      C,HA2
8161 FE 00                       CP      0
8163 28 3A                       JR      Z,HA2
8165 FE 09                       CP      9
8167 F2 9F 81                    JP      P,HA2
816A                    ;
816A F5                          PUSH    AF
816B 21 6F 82                    LD      HL,KEY
816E BE                          CP      (HL)
816F C4 09 81                    CALL    NZ,HRES
8172 F1                          POP     AF
8173                    ;
8173 32 6F 82                    LD      (KEY),A
8176 CD F6 80                    CALL    HSET
8179                    ;
8179 3A 6F 82                    LD      A,(KEY)
817C 21 70 82                    LD      HL,POS
817F BE                          CP      (HL)
8180 20 20                       JR      NZ,HA3
8182                    ;
8182 CD E4 80                    CALL    MOGHIT
8185 3A 6E 82                    LD      A,(SCORE)
8188 3C                          INC     A
8189 27                          DAA
818A 32 6E 82                    LD      (SCORE),A
818D CD C4 1F                    CALL    #BELL
8190 CD 6A 80                    CALL    SCPRT
8193                    ;
8193 01 00 00                    LD      BC,0000H
8196 10 FE              HW:      DJNZ    HW
8198 0D                          DEC     C
8199 20 FB                       JR      NZ,HW
819B CD 09 81                    CALL    HRES
819E C9                          RET
819F                    ;
819F CD 09 81           HA2:     CALL    HRES
81A2 21 6D 82           HA3:     LD      HL,WTIME
81A5 35                          DEC     (HL)
81A6 20 AD                       JR      NZ,HA1
81A8 C9                          RET
81A9                    ;
81A9 3A 71 82           RND16:   LD      A,(RND16B)
81AC 87                          ADD     A,A
81AD F5                          PUSH    AF
81AE F5                          PUSH    AF
81AF E6 80                       AND     80H
81B1 1F                          RRA
81B2 1F                          RRA
81B3 1F                          RRA
81B4 1F                          RRA
81B5 47                          LD      B,A
81B6 F1                          POP     AF
81B7 E6 08                       AND     08H
81B9 A8                          XOR     B
81BA 47                          LD      B,A
81BB F1                          POP     AF
81BC CB 58                       BIT     3,B
81BE 28 04                       JR      Z,RS1
81C0 CB C7                       SET     0,A
81C2 18 02                       JR      RRET
81C4 CB 87              RS1:     RES     0,A
81C6 32 71 82           RRET:    LD      (RND16B),A
```

▶ゼミの学生にパソコンに慣れてもらおうと，麻雀ゲームを研究室に持ち込んだところ，あっという間に研究室はゲーム喫茶になってしまいました。その後半年ほど経ちましたが，テンキー以外をまともに打てる学生はまだいません。　小宮山　政敏（30）千葉県

```
81C9 E6 0F                   AND     0FH
81CB C9                      RET
81CC                 ;
81CC 7B 7B 31 7B     LIN:    DEFM    '■1■2■3■4■'
81D0 7B 7B 7B 32
81D4 7B 7B 7B 7B
81D8 33 7B 7B 7B
81DC 7B 34 7B 7B
81E0 7B 7B 35 7B             DEFM    '■5■6■7■8■'
81E4 7B 7B 7B 36
81E8 7B 7B 7B 7B
81EC 37 7B 7B 7B
81F0 7B 38 7B 7B
81F4 00                      DEFB    00H
81F5                 ;
81F5 20 20 20        MOGDT1: DEFM    '   '
81F8 1F 1D 1D 1D             DEFB    DWN,LFT,LFT,LFT
81FC 20 20 20        MOGDT2: DEFM    '   '
81FF 1F 1D 1D 1D             DEFB    DWN,LFT,LFT,LFT
8203 20 20 20        MOGDT3: DEFM    '   '
8206 1F 1D 1D 1D             DEFB    DWN,LFT,LFT,LFT
820A 20 2D 20        MOGDT4: DEFM    ' - '
820D 1F 1D 1D 1D             DEFB    DWN,LFT,LFT,LFT
8211 4F 20 4F                DEFM    'O O'
8214 1F 1D 1D 1D             DEFB    DWN,LFT,LFT,LFT
8218 3D 2B 3D                DEFM    '=+='
821B                 ;
821B 20 2D 20        MOGDT5: DEFM    ' - '
821E 1F 1D 1D 1D             DEFB    DWN,LFT,LFT,LFT
8222 2A 20 2A                DEFM    '* *'
8225 1F 1D 1D 1D             DEFB    DWN,LFT,LFT,LFT
8229 3E 2B 3C                DEFM    '>+<'
822C 00                      DEFB    00H
822D                 ;
822D 7B              HDAT1:  DEFM    '■'
822E 1F 1D                   DEFB    DWN,LFT
8230 7B 7B 7B                DEFM    '■■■'
8233 1F 1D 1D 1D             DEFB    DWN,LFT,LFT,LFT
8237 7B                      DEFM    '■'
8238 00                      DEFB    00H
8239                 ;
8239 20              HDAT2:  DEFM    ' '
823A 1F 1D                   DEFB    DWN,LFT
823C 20 20 20                DEFM    '   '
823F 1F 1D 1D 1D             DEFB    DWN,LFT,LFT,LFT
8243 20                      DEFM    ' '
8244 00                      DEFB    00H
8245                 ;
8245 53 43 4F 52     SCDAT:  DEFM    'SCORE='
8249 45 3D
824B 00                      DEFB    00H
824C                 ;
824C 2A 2A 20 47     GVDAT:  DEFM    '** GAME OVER **'
8250 41 4D 45 20
8254 4F 56 45 52
8258 20 2A 2A
825B 00                      DEFB    00H
825C 48 49 54 20     STDAT:  DEFM    'HIT SPACE KEY !'
8260 53 50 41 43
8264 45 20 4B 45
8268 59 20 21
826B 00                      DEFB    00H
826C                 ;
826C                 TIME:   DEFS    1
826D                 WTIME:  DEFS    1
826E                 SCORE:  DEFS    1
826F                 KEY:    DEFS    1
8270                 POS:    DEFS    1
8271 01              RND16B: DEFB    01H
```

リスト2 ボールマン（ソースプログラム）

```
0000                 ;
0000                 ;The BALLMAN      by T.Mounai
0000                 ;
1FF4 P               #PRINT  EQU     1FF4H
1FE5 P               #MSX    EQU     1FE5H
1FD0 P               #GETKY  EQU     1FD0H
1FCD P               #BRKEY  EQU     1FCDH
1FC4 P               #BELL   EQU     1FC4H
1FC1 P               #PRTHX  EQU     1FC1H
1FBE P               #PRTHL  EQU     1FBEH
201E P               #LOC    EQU     201EH
2030 P               #WIDCH  EQU     2030H
0000                 ;
001C P               RIT     EQU     1CH
001D P               LFT     EQU     1DH
001F P               DWN     EQU     1FH
0000                 ;
8000                         ORG     8000H
8000                 ;
8000 CD 35 80        MAIN:   CALL    INIT
8003 CD AC 80        REPEAT: CALL    MANMVE
8006 CD 40 81                CALL    BTOS
8009 CD 6D 81                CALL    BSEL
800C                 ;
800C 3A 8F 83                LD      A,(RESULT)
800F FE 01                   CP      1
8011 28 0B                   JR      Z,ERR
8013 FE 02                   CP      2
8015 CC 2D 80                CALL    Z,SCUP
8018 CD CD 1F                CALL    #BRKEY
801B 20 E6                   JR      NZ,REPEAT
801D C9                      RET
801E                 ;
801E CD 9D 82        ERR:    CALL    GOVER
8021 CD D0 1F        ER1:    CALL    #GETKY
8024 FE 20                   CP      20H
8026 28 D8                   JR      Z,MAIN
8028 FE 1B                   CP      1BH
802A 20 F5                   JR      NZ,ER1
802C C9                      RET
802D                 ;
802D AF              SCUP:   XOR     A
802E 32 8F 83                LD      (RESULT),A
8031 CD 64 82                CALL    SCSET
8034 C9                      RET
8035                 ;
8035 3E 28           INIT:   LD      A,40
8037 CD 30 20                CALL    #WIDCH
803A 3E 0C                   LD      A,0CH
803C CD F4 1F                CALL    #PRINT
803F 21 00 0B                LD      HL,0B00H
8042 CD 1E 20                CALL    #LOC
8045 11 AA 82                LD      DE,SCREEN
8048 CD E5 1F                CALL    #MSX
804B                 ;
804B 21 95 83                LD      HL,BPOS
804E 36 15                   LD      (HL),15H
8050 23                      INC     HL
8051 11 97 83                LD      DE,BPOS+2
8054 01 05 00                LD      BC,5
8057 36 FF                   LD      (HL),0FFH
8059 ED B0                   LDIR
805B                 ;
805B AF                      XOR     A
805C 32 9C 83                LD      (BCUNT),A
805F 32 91 83                LD      (KEY),A
8062 32 8F 83                LD      (RESULT),A
8065 32 94 83                LD      (BNUM),A
8068 32 95 83                LD      (BPOS),A
806B 32 92 83                LD      (TIME),A
806E 32 8E 83                LD      (SCORE),A
8071 CD 70 82                CALL    SCPRT
8074                 ;
8074 3E FF                   LD      A,0FFH
8076 32 93 83                LD      (BTIME),A
8079                 ;
8079 3E 02                   LD      A,2
807B 32 90 83                LD      (POS),A
807E CD F0 80                CALL    MANSET1
8081 C9                      RET
8082                 ;
8082 D5              VAR:    PUSH    DE
8083 5F                      LD      E,A
8084 16 00                   LD      D,0
8086 19                      ADD     HL,DE
8087 D1                      POP     DE
8088 C9                      RET
8089                 ;
8089 3A 8D 83        RND16:  LD      A,(RND16B)
808C 87                      ADD     A,A
808D F5                      PUSH    AF
808E F5                      PUSH    AF
808F E6 80                   AND     80H
8091 1F                      RRA
8092 1F                      RRA
8093 1F                      RRA
8094 1F                      RRA
8095 47                      LD      B,A
8096 F1                      POP     AF
8097 E6 08                   AND     08H
8099 A8                      XOR     B
809A 47                      LD      B,A
809B F1                      POP     AF
809C CB 58                   BIT     3,B
809E 28 04                   JR      Z,RS1
80A0 CB C7                   SET     0,A
80A2 18 02                   JR      RRET
80A4 CB 87           RS1:    RES     0,A
80A6 32 8D 83        RRET:   LD      (RND16B),A
80A9 E6 0F                   AND     0FH
80AB C9                      RET
80AC                 ;
80AC CD 35 81        MANMVE: CALL    INKEY
80AF FE 34                   CP      '4'
80B1 CC CE 80                CALL    Z,LEFT
80B4 FE 5A                   CP      'Z'
80B6 CC CE 80                CALL    Z,LEFT
80B9 FE 1D                   CP      LFT
80BB CC CE 80                CALL    Z,LEFT
80BE FE 36                   CP      '6'
80C0 CC DF 80                CALL    Z,RIGHT
80C3 FE 43                   CP      'C'
80C5 CC DF 80                CALL    Z,RIGHT
80C8 FE 1C                   CP      RIT
80CA CC DF 80                CALL    Z,RIGHT
80CD C9                      RET
80CE                 ;
80CE 3A 90 83        LEFT:   LD      A,(POS)
80D1 FE 00                   CP      0
80D3 C8                      RET     Z
80D4 CD 0A 81                CALL    MANSET3
80D7 21 90 83                LD      HL,POS
80DA 35                      DEC     (HL)
80DB CD F0 80                CALL    MANSET1
80DE C9                      RET
80DF                 ;
80DF 3A 90 83        RIGHT:  LD      A,(POS)
80E2 FE 02                   CP      2
80E4 C8                      RET     Z
80E5 CD 0A 81                CALL    MANSET3
80E8 21 90 83                LD      HL,POS
80EB 34                      INC     (HL)
80EC CD F0 80                CALL    MANSET1
80EF C9                      RET
80F0                 ;
80F0 3A 90 83        MANSET1:LD      A,(POS)
80F3 CD 17 81                CALL    MLOC
80F6 11 1C 83                LD      DE,MAN1
80F9 CD E5 1F                CALL    #MSX
80FC C9                      RET
80FD                 ;
80FD 3A 90 83        MANSET2:LD      A,(POS)
8100 CD 17 81                CALL    MLOC
8103 11 2E 83                LD      DE,MAN2
8106 CD E5 1F                CALL    #MSX
8109 C9                      RET
810A                 ;
810A 3A 90 83        MANSET3:LD      A,(POS)
810D CD 17 81                CALL    MLOC
8110 11 40 83                LD      DE,MAN3
8113 CD E5 1F                CALL    #MSX
8116 C9                      RET
8117                 ;
8117 FE 00           MLOC:   CP      0
8119 CC 2C 81                CALL    Z,ML
811C FE 01                   CP      1
811E CC 2F 81                CALL    Z,MM
8121 FE 02                   CP      2
8123 CC 32 81                CALL    Z,MR
8126 26 0D                   LD      H,13
```

▶ひとり言です。全国の数学教員に告ぐ。とーとー，数学の分野にもコンピュータが導入されることになりました。いまからでも遅くない。BASICくらいはやっておいたほうがえーよー。

岩腰　清（34）岐阜県

```
8128 CD 1E 20             CALL    #LOC
812B C9                   RET
812C 2E 04       ML:      LD      L,4
812E C9                   RET
812F 2E 08       MM:      LD      L,8
8131 C9                   RET
8132 2E 0C       MR:      LD      L,12
8134 C9                   RET
8135             ;
8135 CD D0 1F    INKEY:   CALL    #GETKY
8138 21 91 83             LD      HL,KEY
813B BE                   CP      (HL)
813C 77                   LD      (HL),A
813D C0                   RET     NZ
813E AF                   XOR     A
813F C9                   RET
8140             ;
8140 3A 90 83    BTOS:    LD      A,(POS)
8143 FE 00                CP      0
8145 20 02                JR      NZ,BT1
8147 0E 0E                LD      C,0EH
8149 FE 01       BT1:     CP      1
814B 20 02                JR      NZ,BT2
814D 0E 09                LD      C,09H
814F FE 02       BT2:     CP      2
8151 20 02                JR      NZ,BT3
8153 0E 04                LD      C,4
8155             ;
8155 21 95 83    BT3:     LD      HL,BPOS
8158 06 07                LD      B,7
815A 7E          BT4:     LD      A,(HL)
815B B9                   CP      C
815C 28 04                JR      Z,BT5
815E 23                   INC     HL
815F 10 F9                DJNZ    BT4
8161 C9                   RET
8162             ;
8162 34          BT5:     INC     (HL)
8163 CD FD 80             CALL    MANSET2
8166 CD C4 1F             CALL    #BELL
8169 CD 12 82             CALL    BPRT
816C C9                   RET
816D             ;
816D 21 9C 83    BSEL:    LD      HL,BCUNT
8170 34                   INC     (HL)
8171 3A 93 83             LD      A,(BTIME)
8174 BE                   CP      (HL)
8175 C0                   RET     NZ
8176 36 00                LD      (HL),0
8178             ;
8178 21 92 83             LD      HL,TIME
817B 34                   INC     (HL)
817C 3E C8                LD      A,200
817E BE                   CP      (HL)
817F 20 05                JR      NZ,BS1
8181 36 00                LD      (HL),0
8183 CD 83 82             CALL    BMULT
8186             ;
8186 3A 94 83    BS1:     LD      A,(BNUM)
8189 21 95 83             LD      HL,BPOS
818C CD 82 80             CALL    VAR
818F 7E                   LD      A,(HL)
8190 FE FF                CP      0FFH
8192 20 06                JR      NZ,BS2
8194             ;
8194 AF                   XOR     A
8195 32 94 83             LD      (BNUM),A
8198 18 EC                JR      BS1
819A             ;
819A FE 15       BS2:     CP      15H
819C 20 1D                JR      NZ,BS3
819E             ;
819E CD 89 80             CALL    RND16
81A1 FE 0A                CP      10 ·
81A3 FA DF 81             JP      M,INCNUM
81A6             ;
81A6 CD 4E 82             CALL    BCHK
81A9 FE 00                CP      0
81AB C2 DF 81             JP      NZ,INCNUM
81AE             ;
81AE 21 95 83             LD      HL,BPOS
81B1 3A 94 83             LD      A,(BNUM)
81B4 CD 82 80             CALL    VAR
81B7 36 00                LD      (HL),0
81B9 18 20                JR      BS4
81BB             ;
81BB 21 95 83    BS3:     LD      HL,BPOS
81BE 3A 94 83             LD      A,(BNUM)
81C1 CD 82 80             CALL    VAR
81C4 7E                   LD      A,(HL)
81C5             ;
81C5 FE 04                CP      04H
81C7 28 23                JR      Z,MISS
81C9 FE 09                CP      09H
81CB 28 1F                JR      Z,MISS
81CD FE 0E                CP      0EH
81CF 28 1B                JR      Z,MISS
81D1             ;
81D1 3C                   INC     A
81D2 77                   LD      (HL),A
81D3 FE 15                CP      15H
81D5 CC 0C 82             CALL    Z,CUPIN
81D8 CD DF 81             CALL    INCNUM
81DB CD 12 82    BS4:     CALL    BPRT
81DE C9                   RET
81DF             ;
81DF 3A 94 83    INCNUM:  LD      A,(BNUM)
81E2 3C                   INC     A
81E3 FE 08                CP      8
81E5 20 01                JR      NZ,IN1
81E7 AF                   XOR     A
81E8 32 94 83    IN1:     LD      (BNUM),A
81EB C9                   RET
81EC             ;
81EC CD 3F 82    MISS:    CALL    BLOC
81EF 3E 20                LD      A,20H
81F1 CD F4 1F             CALL    #PRINT
81F4 24                   INC     H
81F5 24                   INC     H
81F6 24                   INC     H
81F7 2D                   DEC     L
81F8 CD 1E 20             CALL    #LOC
81FB 3E 6F                LD      A,'o'
81FD CD F4 1F             CALL    #PRINT
8200 CD C4 1F             CALL    #BELL
8203 CD C4 1F             CALL    #BELL
8206 3E 01                LD      A,1
8208 32 8F 83             LD      (RESULT),A
820B C9                   RET
820C             ;
820C 3E 02       CUPIN:   LD      A,2
820E 32 8F 83             LD      (RESULT),A
8211 C9                   RET
8212             ;
8212 06 00       BPRT:    LD      B,0
8214 78          BP1:     LD      A,B
8215 CD 3F 82             CALL    BLOC
8218 3E 20                LD      A,20H
821A CD F4 1F             CALL    #PRINT
821D 04                   INC     B
821E 78                   LD      A,B
821F FE 15                CP      15H
8221 20 F1                JR      NZ,BP1
8223             ;
8223 06 07                LD      B,7
8225 21 95 83             LD      HL,BPOS
8228 7E          BP2:     LD      A,(HL)
8229 FE FF                CP      0FFH
822B 28 0E                JR      Z,BP3
822D FE 15                CP      15H
822F 28 0A                JR      Z,BP3
8231 E5                   PUSH    HL
8232 CD 3F 82             CALL    BLOC
8235 3E 6F                LD      A,'o'
8237 CD F4 1F             CALL    #PRINT
823A E1                   POP     HL
823B 23          BP3:     INC     HL
823C 10 EA                DJNZ    BP2
823E C9                   RET
823F             ;
823F 87          BLOC:    ADD     A,A
8240 21 63 83             LD      HL,BXY
8243 CD 82 80             CALL    VAR
8246 5E                   LD      E,(HL)
8247 23                   INC     HL
8248 56                   LD      D,(HL)
8249 EB                   EX      DE,HL
824A CD 1E 20             CALL    #LOC
824D C9                   RET
824E             ;
824E 21 95 83    BCHK:    LD      HL,BPOS
8251 06 07                LD      B,7
8253 7E          BC1:     LD      A,(HL)
8254 23                   INC     HL
8255 FE 00                CP      0
8257 28 08                JR      Z,BC2
8259 FE 01                CP      1
825B 28 04                JR      Z,BC2
825D 10 F4                DJNZ    BC1
825F AF                   XOR     A
8260 C9                   RET
8261 3E 01       BC2:     LD      A,1
8263 C9                   RET
8264             ;
8264 3A 8E 83    SCSET:   LD      A,(SCORE)
8267 3C                   INC     A
8268 27                   DAA
8269 32 8E 83             LD      (SCORE),A
826C CD 70 82             CALL    SCPRT
826F C9                   RET
8270             ;
8270 21 15 0A    SCPRT:   LD      HL,0A15H
8273 CD 1E 20             CALL    #LOC
8276 11 52 83             LD      DE,SMES
8279 CD E5 1F             CALL    #MSX
827C 3A 8E 83             LD      A,(SCORE)
827F CD C1 1F             CALL    #PRTHX
8282 C9                   RET
8283             ;
8283 21 95 83    BMULT:   LD      HL,BPOS
8286 06 07                LD      B,7
8288 7E          BM1:     LD      A,(HL)
8289 FE FF                CP      0FFH
828B 28 0D                JR      Z,BM2
828D 23                   INC     HL
828E 10 F8                DJNZ    BM1
8290             ;
8290 3E 01                LD      A,1
8292 21 93 83             LD      HL,BTIME
8295 BE                   CP      (HL)
8296 C8                   RET     Z
8297 CB 3E                SRL     (HL)
8299 C9                   RET
829A             ;
829A 36 15       BM2:     LD      (HL),15H
829C C9                   RET
829D             ;
829D 21 15 0C    GOVER:   LD      HL,0C15H
82A0 CD 1E 20             CALL    #LOC
82A3 11 59 83             LD      DE,GVMES
82A6 CD E5 1F             CALL    #MSX
82A9 C9                   RET
82AA             ;
82AA 7B 20 7B 20 SCREEN:  DEFM    '■ ■               '
82AE 20 20 20 20
82B2 20 20 20 20
82B6 20 20 20 20
82BA 20 20
82BC 0D                   DEFB    0DH
82BD 7B 7B 7B 20          DEFM    '■■■               '
82C1 20 20 20 20
82C5 20 20 20 20
82C9 20 20 20 20
82CD 20 20
82CF 0D                   DEFB    0DH
82D0 20 7B 20 20          DEFM    ' ■               O'
82D4 20 20 20 20
82D8 20 20 20 20
82DC 20 20 20 20
82E0 20 4F
82E2 0D                   DEFB    0DH
82E3 20 7B 20 20          DEFM    ' ■              /■'
82E7 20 20 20 20
82EB 20 20 20 20
82EF 20 20 20 20
82F3 2F 7B
82F5 0D                   DEFB    0DH
82F6 20 7B 20 20          DEFM    ' ■               M'
82FA 20 20 20 20
82FE 20 20 20 20
8302 20 20 20 20
8306 20 4D
```


▶世間を騒がせているOS-9/X68000ですが，気になる価格のほうはどうなるのでしょうか。某MS-DOSのように10万円もしたら，ちょっと手が出ません。どうかひとりでも多くOS-9に触れられるように，低価格にしてほしいものです。　佐藤　幸和（19）東京都

```
8308 0D                    DEFB    0DH
8309 7B 7B 7B 7B           DEFM    '[redacted]'
830D 7B 7B 7B 7B
8311 7B 7B 7B 7B
8315 7B 7B 7B 7B
8319 7B 7B
831B 00                    DEFB    00H
831C                 ;
831C 20 4F 20        MAN1: DEFM    ' O '
831F 1F 1D 1D 1D           DEFB    DWN,LFT,LFT,LFT
8323 28 7B 29              DEFM    '(■)'
8326 1F 1D 1D 1D           DEFB    DWN,LFT,LFT,LFT
832A 20 4D 20              DEFM    ' M '
832D 00                    DEFB    00H
832E                 ;
832E 28 4F 29        MAN2: DEFM    '(O)'
8331 1F 1D 1D 1D           DEFB    DWN,LFT,LFT,LFT
8335 20 7B 20              DEFM    ' ■ '
8338 1F 1D 1D 1D           DEFB    DWN,LFT,LFT,LFT
833C 20 4D 20              DEFM    ' M '
833F 00                    DEFB    00H
8340                 ;
8340 20 20 20        MAN3: DEFM    '   '
8343 1F 1D 1D 1D           DEFB    DWN,LFT,LFT,LFT
8347 20 20 20              DEFM    '   '
834A 1F 1D 1D 1D           DEFB    DWN,LFT,LFT,LFT
834E 20 20 20              DEFM    '   '
8351 00                    DEFB    00H
8352                 ;
8352 53 43 4F 52     SMES: DEFM    'SCORE='
8356 45 3D
8358 00                    DEFB    00H
8359                 ;
8359 47 41 4D 45     GVMES: DEFM   'GAME OVER'
835D 20 4F 56 45
8361 52
8362 00                    DEFB    00H
8363                 ;
8363 10 0D           BXY:  DEFW    0D10H
8365 10 0C                 DEFW    0C10H
8367 0F 0B                 DEFW    0B0FH
8369 0E 0B                 DEFW    0B0EH
836B 0D 0C                 DEFW    0C0DH
836D 0D 0B                 DEFW    0B0DH
836F 0C 0A                 DEFW    0A0CH
8371 0B 0A                 DEFW    0A0BH
8373 0A 0B                 DEFW    0B0AH
8375 09 0C                 DEFW    0C09H
8377 09 0B                 DEFW    0B09H
8379 08 0A                 DEFW    0A08H
837B 07 0A                 DEFW    0A07H
837D 06 0B                 DEFW    0B06H
837F 05 0C                 DEFW    0C05H
8381 05 0B                 DEFW    0B05H
8383 04 0A                 DEFW    0A04H
8385 03 09                 DEFW    0903H
8387 02 09                 DEFW    0902H
8389 01 0A                 DEFW    0A01H
838B 01 0B                 DEFW    0B01H
838D                 ;
838D 01              RND16B: DEFB  01H
838E                 SCORE:  DEFS  1
838F                 RESULT: DEFS  1
8390                 POS:    DEFS  1
8391                 KEY:    DEFS  1
8392                 TIME:   DEFS  1
8393                 BTIME:  DEFS  1
8394                 BNUM:   DEFS  1
8395                 BPOS:   DEFS  7
839C                 BCUNT:  DEFS  1
```

リスト3　もぐらたたき（オブジェクト）

```
8000 3E 28 CD 30 20 CD 47 80 : 17
8008 21 00 11 CD 1E 20 11 5C : AA
8010 82 CD E5 1F CD 7D 80 FE : 1B
8018 1B CA FA 1F FE 20 20 F4 : 30
8020 CD 55 80 CD 84 80 CD 50 : 90
8028 81 CD AE 80 21 6C 82 35 : C0
8030 20 F1 21 00 10 CD 1E 20 : 4D
8038 11 4C 82 CD E5 1F 06 05 : BB
8040 CD C4 1F 10 FB 18 BE 3E : CF
8048 00 32 6F 82 32 6E 82 3E : 83
8050 32 32 6C 82 C9 3E 0C CD : 32
8058 F4 1F 21 00 0D CD 1E 20 : 4C
8060 11 CC 81 CD E5 1F CD 6A : 66
8068 80 C9 21 00 0F CD 1E 20 : 84
8070 11 45 82 CD E5 1F 3A 6E : 51
8078 82 CD C1 1F C9 CD A9 81 : EF
---------------------------------
SUM: 92 0C 8E 22 48 CB A3 5A 78A0

8080 CD D0 1F C9 CD A9 81 CB : 47
8088 3F 3C 32 70 82 CD 1C 81 : 09
8090 26 0A CD 1E 20 11 FC 81 : C9
8098 CD D2 80 CD 1E 20 11 03 : 3E
80A0 82 CD D2 80 CD 1E 20 11 : BD
80A8 0A 82 CD D2 80 C9 3A 70 : 1E
80B0 82 CD 1C 81 26 0A CD 1E : 07
80B8 20 11 03 82 CD D2 80 CD : A2
80C0 1E 20 11 FC 81 CD D2 80 : EB
80C8 CD 1E 20 11 F5 81 CD D2 : 31
80D0 80 C9 06 11 1A 13 CD F4 : 4E
80D8 1F 10 F9 01 50 00 10 FE : 87
80E0 0D 20 FB C9 3A 70 82 CD : EA
80E8 1C 81 26 0A CD 1E 20 11 : E9
80F0 1B 82 CD E5 1F C9 3A 6F : E0
80F8 82 CD 1C 81 2C 26 07 CD : 12
---------------------------------
SUM: 7D 1C 96 D1 FF 48 B0 9A 3501

8100 1E 20 11 2D 82 CD E5 1F : CF
8108 C9 3A 6F 82 CD 1C 81 2C : 8A
8110 26 07 CD 1E 20 11 39 82 : 04
8118 CD E5 1F C9 FE 01 28 1B : DC
8120 FE 02 28 1A FE 03 28 19 : 84
8128 FE 04 28 18 FE 05 28 17 : 84
8130 FE 06 28 16 FE 07 28 15 : 84
8138 2E 24 C9 2E 01 C9 2E 06 : 47
8140 C9 2E 0B C9 2E 10 C9 2E : 00
8148 15 C9 2E 1A C9 2E 1F C9 : 05
8150 3E 40 32 6D 82 CD 7D 80 : 69
8158 FE 00 28 43 CD B8 1F 38 : 45
8160 3E FE 00 28 3A FE 09 F2 : 97
8168 9F 81 F5 21 6F 82 BE C4 : A9
8170 09 81 F1 32 6F 82 CD F6 : 61
8178 80 3A 6F 82 21 70 82 BE : 7C
---------------------------------
SUM: 82 E7 95 9C E7 08 07 4C 10BF

8180 20 20 CD E4 80 3A 6E 82 : 9B
8188 3C 27 32 6E 82 CD C4 1F : 35
8190 CD 6A 80 01 00 00 10 FE : C6
8198 0D 20 FB CD 09 81 C9 CD : 15
81A0 09 81 21 6D 82 35 20 AD : 9C
81A8 C9 3A 71 82 87 F5 F5 E6 : 4D
81B0 80 1F 1F 1F 1F 47 F1 E6 : 1A
81B8 08 A8 47 F1 CB 58 28 04 : 37
81C0 CB C7 18 02 CB 87 32 71 : A1
81C8 82 E6 0F C9 7B 7B 31 7B : E2
81D0 7B 7B 7B 32 7B 7B 7B 7B : 8F
81D8 33 7B 7B 7B 7B 34 7B 7B : 49
81E0 7B 7B 35 7B 7B 7B 7B 36 : 4D
81E8 7B 7B 7B 7B 37 7B 7B 7B : 94
81F0 7B 38 7B 7B 00 20 20 20 : 09
81F8 1F 1D 1D 1D 20 20 20 1F : F5
---------------------------------
SUM: 1B 41 D7 25 0C 38 C8 BB 76AD

8200 1D 1D 1D 20 20 20 1F 1D : F3
8208 1D 1D 20 2D 20 1F 1D 1D : 0[illegible]
8210 1D 4F 20 4F 1F 1D 1D 1D : 51
8218 3D 2B 3D 20 2D 20 1F 1D : 4E
8220 1D 1D 2A 20 2A 1F 1D 1D : 07
8228 1D 3E 2B 3C 00 7B 1F 1D : 79
8230 7B 7B 7B 1F 1D 1D 1D 7B : 62
8238 00 20 1F 1D 20 20 20 1F : DB
8240 1D 1D 1D 20 00 53 43 4F : 5C
8248 52 45 3D 00 2A 2A 20 47 : 8F
8250 41 4D 45 20 4F 56 45 52 : 2F
8258 20 2A 2A 00 48 49 54 20 : 79
8260 53 50 41 43 45 20 4B 45 : 1C
8268 59 20 21 00 08 49 47 20 : 52
8270 20 01                   : 21
---------------------------------
SUM: E5 F4 B4 D7 01 D8 7F B5 D06F
```

リスト4　ボールマン（オブジェクト）

```
8000 CD 35 80 CD AC 80 CD 40 : 88
8008 81 CD 6D 81 3A 8F 83 FE : 86
8010 01 28 0B FE 02 CC 2D 80 : AD
8018 CD CD 1F 20 E6 C9 CD 9D : F2
8020 82 CD D0 1F FE 20 28 D8 : 5C
8028 FE 1B 20 F5 C9 AF 32 8F : 67
8030 83 CD 64 82 C9 3E 28 CD : 32
8038 30 20 3E 0C CD F4 1F 21 : 9B
8040 00 0B CD 1E 20 11 AA 82 : 53
8048 CD E5 1F 21 95 83 36 15 : 55
8050 23 11 97 83 01 05 00 36 : 8A
8058 FF ED B0 AF 32 9C 83 32 : CE
8060 91 83 32 8F 83 32 94 83 : A1
8068 32 95 83 32 92 83 32 8E : 51
8070 83 CD 70 82 3E FF 32 93 : 44
8078 83 3E 02 32 90 83 CD F0 : C5
---------------------------------
SUM: 07 DD 03 F4 F6 11 13 43 2F0F

8080 80 C9 D5 5F 16 00 19 D1 : 7D
8088 C9 3A 8D 83 87 F5 F5 E6 : 6A
8090 80 1F 1F 1F 1F 47 F1 E6 : 1A
8098 08 A8 47 F1 CB 58 28 04 : 37
80A0 CB C7 18 02 CB 87 32 8D : BD
80A8 83 E6 0F C9 CD 35 81 FE : C2
80B0 34 CC CE 80 FE 5A CC CE : 40
80B8 80 FE 1D CC CE 80 FE 36 : E9
80C0 CC DF 80 FE 43 CC DF 80 : 97
80C8 FE 1C CC DF 80 C9 3A 90 : D8
80D0 83 FE 00 C8 CD 0A 81 21 : C2
80D8 90 83 35 CD F0 80 C9 3A : 88
80E0 90 83 FE 02 C8 CD 0A 81 : 33
80E8 21 90 83 34 CD F0 80 C9 : 6E
80F0 3A 90 83 CD 17 81 11 1C : DF
80F8 83 CD E5 1F C9 3A 90 83 : 6A
---------------------------------
SUM: 1E 2D 44 9D E0 C1 32 84 60A0

8100 CD 17 81 11 2E 83 CD E5 : D9
8108 1F C9 3A 90 83 CD 17 81 : 9A
8110 11 40 83 CD E5 1F C9 FE : 6C
8118 00 CC 2C 81 FE 01 CC 2F : 73
8120 81 FE 02 CC 32 81 26 0D : 33
8128 CD 1E 20 C9 2E 04 C9 2E : FD
8130 08 C9 2E 0C C9 CD D0 1F : 90
8138 21 91 83 BE 77 C0 AF C9 : A2
8140 3A 90 83 FE 00 20 02 0E : 7B
8148 0E FE 01 20 02 0E 09 FE : 44
8150 02 20 02 0E 04 21 95 83 : 6F
8158 06 07 7E B9 28 04 23 10 : A3
8160 F9 C9 34 CD FD 80 CD C4 : D1
8168 1F CD 12 82 C9 21 9C 83 : 89
8170 34 3A 93 83 BE C0 36 00 : 38
8178 21 92 83 34 3E C8 BE 20 : 4E
---------------------------------
SUM: 31 79 9D 39 24 FE 07 BC 0FDC

8180 05 36 00 CD 83 82 3A 94 : DB
8188 83 21 95 83 CD 82 80 7E : 09
8190 FE FF 20 06 AF 32 94 83 : 1B
8198 18 EC FE 15 20 1D CD 89 : AA
81A0 80 FE 0A FA DF 81 CD 4E : FD
81A8 82 FE 00 C2 DF 81 21 95 : 58
81B0 83 3A 94 83 CD 82 80 36 : D9
81B8 00 18 20 21 95 83 3A 94 : 3F
81C0 83 CD 82 80 7E FE 04 28 : FA
81C8 23 FE 09 28 1F FE 0E 28 : A5
81D0 1B 3C 77 FE 15 CC 0C 82 : 3B
81D8 CD DF 81 CD 12 82 C9 3A : 91
81E0 94 83 3C FE 08 20 01 AF : 29
81E8 32 94 83 C9 CD 3F 82 3E : DE
81F0 20 CD F4 1F 24 24 24 2D : 99
81F8 CD 1E 20 3E 6F CD F4 1F : 98
---------------------------------
SUM: 64 78 C7 62 6B F4 45 10 789D

8200 CD C4 1F CD C4 1F 3E 01 : 9F
8208 32 8F 83 C9 3E 02 32 8F : 0E
8210 83 C9 06 00 78 CD 3F 82 : 58
8218 3E 20 CD F4 1F 04 78 FE : B8
8220 15 20 F1 06 07 21 95 83 : 6C
8228 7E FE FF 28 0E FE 15 28 : EC
8230 0A E5 CD 3F 82 3E 6F CD : F7
8238 F4 1F E1 23 10 EA C9 87 : 61
8240 21 63 83 CD 82 80 5E 23 : 57
8248 56 EB CD 1E 20 C9 21 95 : CB
8250 83 06 07 7E 23 FE 00 28 : 57
8258 08 FE 01 28 04 10 F4 AF : E6
8260 C9 3E 01 C9 3A 8E 83 3C : 58
8268 27 32 8E 83 CD 70 82 C9 : F2
8270 21 15 0A CD 1E 20 11 52 : AE
8278 83 CD E5 1F 3A 8E 83 CD : 6C
---------------------------------
SUM: E7 02 E9 E3 68 3C 15 C2 F611

8280 C1 1F C9 21 95 83 06 07 : EF
8288 7E FE FF 28 0D 23 10 F8 : DB
8290 3E 01 21 93 83 BE C8 CB : C7
8298 3E C9 36 15 C9 21 15 0C : 5D
82A0 CD 1E 20 11 59 83 CD E5 : AA
82A8 1F C9 7B 20 7B 20 20 20 : 5E
82B0 20 20 20 20 20 20 20 20 : 00
82B8 20 20 20 20 0D 7B 7B 7B : FE
82C0 20 20 20 20 20 20 20 20 : 00
82C8 20 20 20 20 20 20 20 0D : ED
82D0 20 7B 20 20 20 20 20 20 : 5B
82D8 20 20 20 20 20 20 20 20 : 00
82E0 20 4F 0D 20 7B 20 20 20 : 77
82E8 20 20 20 20 20 20 20 20 : 00
82F0 20 20 20 2F 7B 0D 20 7B : B2
82F8 20 20 20 20 20 20 20 20 : 00
---------------------------------
SUM: E7 98 E7 71 A5 B0 7B BE 7B06

8300 20 20 20 20 20 20 20 4D : 2D
8308 0D 7B 7B 7B 7B 7B 7B 7B : 6A
8310 7B 7B 7B 7B 7B 7B 7B 7B : D8
8318 7B 7B 7B 00 20 4F 20 1F : 1F
8320 1D 1D 1D 28 7B 29 1F 1D : 5F
8328 1D 1D 20 4D 20 00 28 4F : 3E
8330 29 1F 1D 1D 1D 20 7B 20 : 5A
8338 1F 1D 1D 1D 20 4D 20 00 : 03
8340 20 20 20 1F 1D 1D 1D 20 : F6
8348 20 20 1F 1D 1D 1D 20 20 : F6
8350 20 00 53 43 4F 52 45 3D : D9
8358 00 47 41 4D 45 20 4F 56 : DF
8360 45 52 00 10 0D 10 0C 0F : DF
8368 0B 0E 0B 0D 0C 0D 0B 0C : 61
8370 0A 0B 0A 0A 0B 09 0C 09 : 52
8378 0B 08 0A 07 0A 06 0B 05 : 44
---------------------------------
SUM: 6A 01 FA BF 0A D3 17 EA 44EE

8380 0C 05 0B 04 0A 03 09 02 : 38
8388 09 01 0A 01 0B 01 25 01 : 47
8390 01 00 85 FF 00 0E 06 05 : 9E
8398 FF FF FF FF 00          : FC
---------------------------------
SUM: 15 05 99 03 15 12 34 08 F0DB
```

▶スゴロクが載っているなんて，どこかのマンガ雑誌みたいですね。「X68000現象を探る」は面白く読ませていただきましたが，多少，偏見があるような気が……。

上野　貢市（21）埼玉県

割り込みってなんだろな

BASICやほかの高級言語ではちょっとできない処理，それが割り込み処理です。ここではX1用のCTCを使ったタイマ割り込みをプログラミングしてみましょう。これぞ，マシン語の醍醐味といえます。

Nishikawa Zenji 西川 善司

超入門Z80マシン語活用術

愛，おぼえてますか？　西川善司です(ちょっと古かったかな)。皆さん「割り込み」って知っていますか。「割り込み」は私の得意技なんです。この間もバスに乗るために並ぼうとしましたが，列が非常に長い。バスがやってきたので，列の前のほうに行き「おぉ，若林じゃないか！　こんなところで会うとは奇遇だねぇ。はっはっはっ」と見知らぬ人の肩を叩き乗り込んでしまいました。

こんなことが，みんなの使っているパソコンのなかでも起こっているんです（吉田:ちょっと違うぞ)。名前ぐらいは聞いたことがあるでしょう。割り込み処理はマシン語ならではの醍醐味を持ったテクニックです。今回はZ80の割り込みについて解説したいと思います。話によってはX1に限定されることをご了承ください。

X1ではモード2

割り込みとは，すごーく簡単にいうとCPUがある仕事をしている最中に，周辺のLSIなどが，

「CPUさん，お願い，あたしをかわいがってよ」

といい，CPUが，

「ちっ，しょーがねーなぁ。ちっとだけだぜ」

と，いまやっている仕事を一時中断して別の仕事をすることだと思ってください（吉田：詳しい人が読んだら怒るぞ，もう)。

Z80には大きく分けて，2種類の割り込みがあります。

NMI (NON-MASKABLE INTERRUPT)
INT (MASKABLE INTERRUPT)

の2つです。

NMIは，CPUがどんな仕事をしていても絶対に禁止できない割り込みです（Z80のNMI端子を0レベルにすると，この割り込みがかかる)。INTは，プログラムによって割り込みを許可，禁止を設定できる，パソコンのプログラミングにおいても使用頻度の高いものです（Z80のINT端子を0レベルにすると，この割り込みがかかる)。

INTには，

IM 0	機械語コード	ED	46
IM 1		ED	56
IM 2		ED	5E

の3つのモードがあります。X1の割り込みはほとんどIM2ですのでこれを重点的に説明してみましょう。

実際，IM2はX1でもっともよく使われる割り込み方法です。キーボード然り，FM音源然り，CTC然り，ありをりはべりいまそかり。せ，まる，き，し，しか，まる。おっと，過去助動詞「き」の活用が出てしまった。

このモードでは周辺LSIが割り込み要求をすると割り込みベクトルという，1バイトのデータをCPUに渡します。するとCPUはIレジスタという，割り込み処理用の1バイトのレジスタを上位とし，送られてきた1バイトのベクトルを下位としたアドレスを参照します。このアドレスに書かれている2バイトが示すアドレスにジャンプします。それから，CPUはRETI（機械語コード：ED 4D）を見つけるまで割り込みルーチンを実行します。C9HのRETでないことに注意。うーん。このモードは少し複雑だから，私お得意の擬人法で説明しなおしてみましょう。

CPU「かりかり（仕事をしている)」

LSI「CPUさん，誰かがキーボードを押したみたいなので，ちょっとお使いにいってほしいんですけど」

C「ガッテンダー。どこまでだい？」

L「ええと，住所は××52Hに書いておいたわ。それを読んでちょうだいね」

そういうとL子は「52H」と書かれた板切れをCPU夫に渡し，どこかにいってしまった。

C「××52Hだぁ？　上位バイトもいってくんなきゃ，どこ探していいかわかんねぇじゃねぇか。……ハッ！　そういえば昔，神様が，俺が目覚めたときに『CPUよ，よく聞け。もし，周辺LSIに仕事を頼まれたらこのIレジスタお札を使うがよい』といっていたっけ。このIレジスタお札とL子のくれた52Hと書かれた板切れはぴったしと合うぞーっ。す，すると……？　Iレジスタには00Hと書かれているぞーっ。そしてL子のくれた板には52H！　よって，0052H番地へ見にいけばいいのかぁ。う〜ま〜い〜ぞーっ!!!」

⋮

C「0052Hというとこのあたりだな。おや，これがL子のいっていたメモだな。なになに，『0346H』か。ここへいって仕事をすりゃいいのね」

⋮

C「ここが0346Hだ。俺は『RETI』があるまでここで仕事をすりゃいいんだな。ブラボー!!」

モード2割り込みの使い方

このモードの利点のひとつは，割り込みルーチンをユーザーが自由に64Kバイトのメモリ中どこにでも置けるということです。では，このモードを使用しているというX1BASICを逆アセンブルしてみましょう。

```
011B:ED 5E      IM 2
011D:21 46 03   LD HL,0346H
0120:22 52 00   LD (0052H),HL
0123:CD 2D 01   CALL 012DH
```

モード0とモード1割り込み

IM 0はZ80のご先祖様の8080というCPUが持っていた割り込みです。これは周辺LSIがCPUに対してRST命令（1バイトのCALL文みたいなもんだね）やCALL命令を発信して割り込みを実行させるモードです。Z80は動作を開始したときはこのモードになっています。

IM1は周辺LSIが割り込み要求する（本文に書いたようにINT端子を0レベルにする）ことによってCALL 0038HをCPUは実行します。先ほどのNMIのときと同様に0038Hからは割り込みルーチンがなければなりません。IM0とIM1とも割り込みルーチンからの帰還は単なる「RET」（機械語コード：C9H）で行います。吉田君の話ではPC-8801などのBGMルーチンはIM1だそうです。X1のFM音源はINT端子とはつながってませんから，IM1でBGMルーチンは作れませんね。よって12月号のMMLはIM2です。

表1　X1/turboのCTCアドレス

チャンネル	turbo	CZ-8BM2	CZ-8BRI	CZ-BSI
0	IFA0H	IFA8H	0A04H	0704H
1	IFA1H	IFA9H	0A05H	0705H
2	IFA2H	IFAAH	0A06H	0706H
3	IFA3H	IFABH	0A07H	0707H

注：CZ-8BM2とはRS-232Cマウスボード
　　CZ-8BR1とは立体映像セット
　　CZ-8BS1とはFM音源ボード

```
0126:CD 3C 01   CALL 013CH
0129:FB         EI
012A:C3 CC 14   JP 14CCH
012D:21 52 00   LD HL,0052H
0130:7C         LD A,H
0131:ED 47      LD I,A
0133:3E E4      LD A,0E4H
0135:CD FE 0D   CALL 0DFEH
0138:7D         LD A,L
0139:C3 54 0B   JP 0B54H
```

となります。011B$_H$に，ほら，IM2がありました。0129$_H$のEIというのがありますね。これも割り込みに関係ある命令なんです。これはEnable Interruptの略，つまり割り込み許可という意味です。逆の命令もあります。DI(Disable Interrupt 機械語コード：F3$_H$）がそうです。これはさっきの，IM0やIM1にも有効な命令で，CPUがとてもタイミングにかかわる大切な仕事をしているときなど，割り込みなどがかかっては困るときなどに使います。Z80の初期状態は，まさにこのDIの状態ですから，割り込み処理があるようなプログラムの先頭などにはEIを書いておく必要があります。

話は戻って0123$_H$にサブルーチンCALLします（吉田：なぜ011D$_H$から0120$_H$の説明を飛ばしたかはすぐわかるよ)。0123$_H$〜0130$_H$ではHL＝0052$_H$，A＝Hを行っています。0131$_H$のLD I,A。Aの内容をIへ入れてますね。Iは，割り込みアドレスの上位バイトとなります。ここでは00$_H$が入りますよね。なぜって，012D$_H$でHL＝0052$_H$ですから，H＝00$_H$，L＝52$_H$。そして，0130$_H$でA＝H＝00$_H$ですから。

0133$_H$のLD A,0E4$_H$でA＝0E4$_H$。0135$_H$で0DFE$_H$にCALLしてますが，0DFE$_H$にはキーボードを操るサブCPUへコマンドを送るためのルーチンがあり，0E4$_H$がそのコマンドとなるわけです。ちなみに0E4$_H$は割り込みベクトルの設定ということです。そして，0138$_H$ではA＝L＝52$_H$。0139$_H$で0B54$_H$にジャンプしています。0B54$_H$にはサブCPUにデータを渡すルーチンがあるのです。つまり！ 0133$_H$以降は，まさに！！ 周辺LSIへの割り込みベクトル設定なのです！

以上のことからIには00$_H$，サブCPU（周辺LSI）には52$_H$が設定されました。もしも，キーボードが押されると0052$_H$に書かれている2バイトのアドレスへジャンプ（正確にはCALLですが）するわけです。そういえば0052$_H$にはなにが書かれているのでしょう。011D$_H$から0120$_H$を見てください。0052$_H$に0346$_H$を書き込んでいますね。つまり，キーボードを押すと0346$_H$へ飛ぶ（キー割り込み）ですね。当然のことながら0346$_H$からはキー処理ルーチンなどがあるわけです。さっきの擬人法説明文はまさにこの例を説明していたので，もう一度読んでみてください。

CTC割り込みのあれこれ

そろそろ実戦的な話をここから始めましょう。ここではCTCの具体的な使い方を講義します。CTCを使えば並列処理（に近いようなこと）やプリンタバッファのように一定間隔で見張っていたいといった要求にも応えることができます。

X1/turboのCTCのアドレスは表1のようになっています(以後本文はturbo用CTCのアドレスを用いて話を進めていくのでノーマルX1の方は自分の持っているCTCのアドレスに置き換えてください)。

CTCを操るには

```
LD    BC,1FA0H
LD    A,n
OUT   (C),A
```

のようにすればよいことになります。BASICからはOUT ＆H1FA0,nですね。

さて，この「n」にはなにを入れたらいいのでしょうか。表2を見てください。CTCのコマンドのようなものを載せておきます。皆さんがいちばん気になると思われるタイマモードとカウンタモードとの違いですが，両者際立った違いはありませんが，タイマモードはSIOに供給されているクロック（X1では4MHz）を数え，カウンタモードではCLK/TRGnという端子にくるパルスを数えます。これについてはのちほど，詳しく説明しましょう。

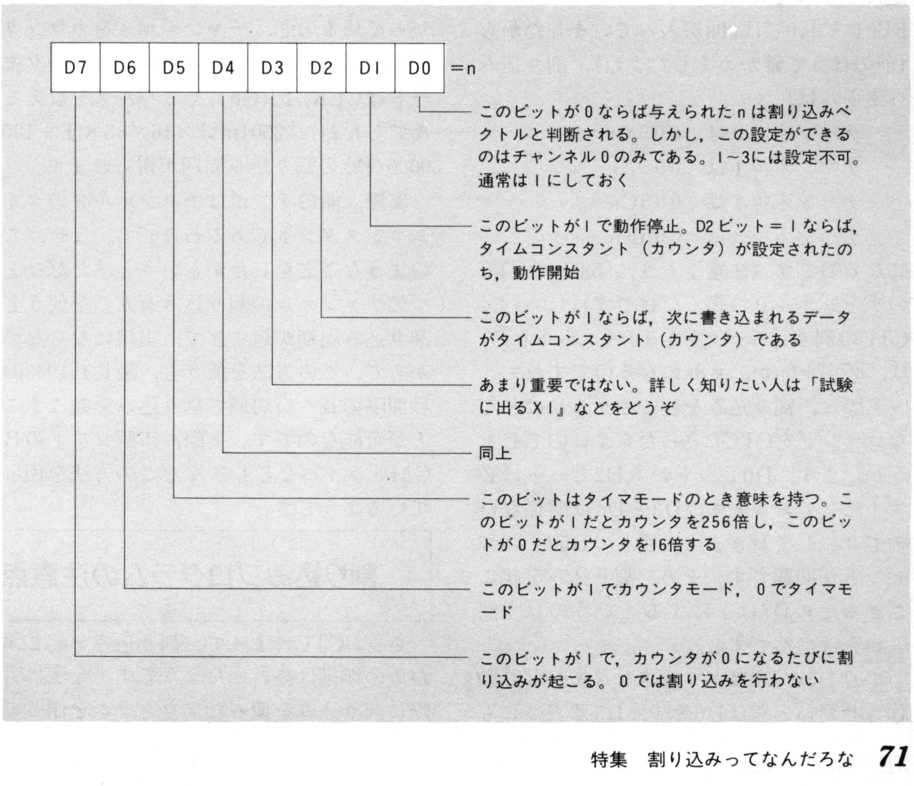
それでは，表2を見て，実際にCTCコマンドを作ってみましょう。とても簡単ですから，よく見ててくださいよ。

まず，リセットコマンドは，

＆B00000011＝3

ですね。なぜかというと，D0は常に1にしなければいけないし，D1が1だとリセットだからです。CTCチャンネル0のポートにこの3を書き込んでやるとそこのCTCチャンネルはリセットされます。表をじっくり見て納得してください。

表2　CTCのコマンド



割り込みベクトルの設定

次に割り込みベクトルはどうやって設定したらよいのでしょう。表2にあるとおりにD0ビットが0である1バイトのデータをCTCのチャンネル0に送ってやればいいというわけではないのです。そもそも，CTCは0～3の4つのチャンネルを持っていて，それぞれ，違った時間に別々の割り込みがかけられるのにどうして割り込みベクトルの設定はチャンネル0にしか行えないのだと，思った方も大勢いると思います。

実は，割り込みベクトルをチャンネル0に設定すると自動的に1～3のチャンネルにも設定が行われるのです。そこで，割り込みベクトルの設定はD0ビットが0でなければならないという制約のほかに「チャンネル0に設定する割り込みベクトルは下3ビットが000でなければならない」という制約もあったのです。

だからチャンネル0に設定すべき割り込みベクトルは，

&B＊＊＊＊＊000

でなければならないということです。そして，チャンネル0に設定されるとその瞬間，

チャンネル1は　＊＊＊＊＊010
チャンネル2は　＊＊＊＊＊100
チャンネル3は　＊＊＊＊＊110

と1から3のチャンネルの割り込みベクトルも設定されるのです。たとえば，12月号のMMLなどは，0チャンネルに58_Hを書き込んでいました。Iレジスタには，CZ-8FB01 V1.0では00_Hが入っていましたから(前のほうで確かめましたよね)，割り込みベクトルは，

チャンネル0は　0058_H
チャンネル1は　$005A_H$
チャンネル2は　$005C_H$
チャンネル3は　$005E_H$

になるのです（注意：そう，「58_H」をCTCのチャンネル0に書くだけです)。よって，CTCの割り込みベクトル（チャンネル0の）は，必ず$*0_H$か，$*8_H$になるのですねぇ。

実際に，割り込みを起こさせるにはどんなコマンドをCTCに送ったらよいのでしょうか。まず，D0ビットからD2ビットは必ず1にしておきます。D3とD4は関係ないので0にしておきます。残りのD5からD7ビットが問題です。まあ，割り込みを起こさせるためD7は1にするというのは当然といえば当然ですね。

ではD5とD6は？　カウンタモードで動作させたいときはD6を1，D5は表2にもあるようにカウンタモードのときには関係ないので0にでもしておきます。すると，

&B11000111＝199

になりますね。タイマモードのときはD6を0，D5はタイムコンスタント（カウンタ）を何倍にするかということですね。実用上は256倍（D5ビットを1）がいいと思います。そもそも，タイマモードでは4MHzという速いパルスを数えるので16倍やそこらでは，割り込み周期はかなり短くなってしまうのです。まあ，そんなわけで，タイマモードのケースでは，

&B10100111＝167

となります。

ところでD2ビットが1になっているので，この，199もしくは167(または135)をCTCに書いてやったあとにタイムコンスタント（カウンタ）も書いてやらなければなりません。これは，自分の好みの値で結構。完全なデータですから，割り込みベクトルの設定のときみたいに変な制約はありません。説明しますと，1がもっとも小さい値で，割り込み間隔がもっとも狭く，大きくすればするほど，割り込み間隔は広くなります(吉田：255がもっとも大きいのではなくて，0がいちばん大きい値となるんだよ。この0とは正確には256のことなんだけどね)。

12月号のMMLでは，CTCをどんなふうに使っていたのでしょう。実は，チャンネル0をタイマモードで割り込みなし，プリスケーラは16倍で使いました（タイムコンスタントは85の固定)。チャンネル0のZC/TO_0はチャンネル3のCLK/TRG_3とつながっているので，チャンネル3をカウンタモード，割り込み有りで使い（カウンタモードはCLK/TRGnにくるパルスを数えるんでしたね)，250(ms)×16×85×T÷1000000(秒)の割り込み周期が得られます。

実際，曲のテンポはチャンネル3のタイムコンスタントであるわけです。なぜ，このようなことをしたかというと,「ただひとつのチャンネルの割り込み有り」を使うと，割り込み周期が短すぎて，実用にならないからで，この方法を使うと，最長4.194304秒間隔の長～い周期で割り込みを起こすことが可能なのです。実際，市販ソフトのBGMドライバなどもみんなこの方法を使っているようです。

割り込みプログラムの注意点

さぁ，CTCによって，割り込みを起こすための準備は終わったようです。今度は実際に割り込みを使ったプログラムを作ってみましょう。



割り込みを使ったプログラムを作るうえでもっとも気をつけなければならないのは，レジスタの保存です。CPUがある仕事をしているときに突然割り込みがかかるとします。そこで割り込みルーチンに飛び，そこでの仕事を終えて戻ってきたとします。割り込みがかかる前の仕事を再び続けるのですが，このとき，割り込みのかかる前とかかったあとでのレジスタの内容がもし違っていたらどうなるでしょう。そう，不都合が起きますよね。たいていの場合，暴走するでしょう。

そんなわけで割り込みルーチンは，（IM2時）次に示すような形で組まれることが多いですね。

```
  1  DI
  2  LD     (INTSP),SP
  3  LD     SP,INTSP
  4  PUSH   AF
  5  PUSH   BC
  6  PUSH   DE
  7  PUSH   HL
          :
100  POP    HL
101  POP    DE
102  POP    BC
103  POP    AF
104  LD     SP,(INTSP)
105  EI
106  RETI
          :
200  DS     32
201 INTSP:
202  DW     0000H
          :
```

1行でDIによって割り込み禁止。これはプログラムによって必要不必要があるでしょうね。割り込み処理中に，割り込みをかけたい場合にはEIを書いておくか，または，前のDIを取っ払いましょう。2行目でスタックをワークにセーブ，そして，新しいスタックを設定しています。この理由は，割り込みのかかったときのスタックが

必ずしもよい条件であるとは限らないためです。ですから割り込みルーチンのほうで，ちゃんとしたスタックポインタを与えているわけです。処理が終わって，レジスタを割り込みのかかる前の状態にしてやります(100行から103行)。

104行ではスタックをも前の状態にしてやります。105行で割り込み許可（再び割り込みのかかるように)。106行で割り込み終了。割り込みのかかる前のアドレスに帰還します。割り込みルーチンで，裏レジスタなども使っている場合はそれらも保存しなければなりませんが，逆に，レジスタをひとつしか使わなければ保存するレジスタはそのひとつだけでいいのです。

そして星は流れる

これらを踏まえて作成したのがリスト1です。これを，X1用 BASIC(turboBASICでは駄目)，またはS-OS上で入力して，セーブしてください。また，ソースリストにも書いておきましたが，ダンプリストはいまX1turbo用になっていますのでノーマルX1の人は，表3に従って変更箇所を変更してからセーブしてください。CZ-8FB01 V1.0もしくはCZ-8CB01 V1.0を起動したのち，

CLEAR &HFBB0

を実行，その後，リスト2でPCGを設定し，さっき入力したオブジェクトをロード，

CALL &HFBB0

を実行してください。

すると，おぉ……，星が6段階のスピードで斜めに流れていくぞぉ。あぁ，カーソルが点滅している。キーを押すとどうだぁー。おぉ，も，文字が出たぁ。ということは，星を流したまま，プログラムが組めるのかぁ。ディスクのディレクトリをとってみるとするかな。おおっ，星が，星が，星が，文字をスキップして流れているぞう。これはまるで宇宙に浮かぶようじゃあ！(星の流れる向きが逆じゃないのかとかはいわないように)。

さて，このプログラム，まだ改善の余地(カーソル行には星を流さないほうがいいと吉田君にいわれた）があるけどさっさっと説明にいってしまう善司でした。

7行のHOWMというのは星の数です。好きに変えてみてください。YMAXは星の現れる最下段のY座標。YTOPは星の消えるY座標。19行から22行はBASICのテキストスクロールへパッチを当てています。テキストをスクロールさせたときに，星がおかしくなるのを防ぐためです。31行から35行まで割り込みベクトルをCTCに設定しています。これで割り込みが起こったときはINT__STARというルーチンへ飛んでいくのです。37行でAレジスタに167を入れていますが，これはもうどんな意味か皆さんはわかっていますよね。39行はタイムコンスタント（カウンタ）です。この値を小さくすると星が速くなり，大きくすると遅くなります。

ROLL__PATCHというルーチンはさっきいったテキストスクロールルーチンへのパッチ処理です。TOROKUは星の初期設定を行うためのルーチンです。データ形式はそこの行にある注釈のとおりです。SET VALというのはその設定を実際に行っているところです。RNDは乱数ルーチンです。これは比較的ちゃんとした（?）乱数を速く出してくれるルーチンなので私は使い回ししております。

INT__STARは実際に，割り込みが起きたときに飛んでくるルーチンです。頭のほうでレジスタを保存しているのがわかりますね。

LD A,HOWM/2というのは1回の割り込みでいくつの星を処理するかを決めています。ここを変えると星のスピードに影響します。いろいろ試してみると面白いでしょう。

星のスピードのもとになっているカウンタをデクリメントし，ゼロにならなければ星を描いたりするルーチンをスキップします。ゼロになったならば，前の割り込みに描いた，星を消します。次に星の座標などを処理します。

FI2ではもし，文字があったら星は描かないというような処理を行なっています。

HANTEI,SETVAL__FOR__RETで，次の星の処理をするかしないかを見ています。その後ろでは，割り込み終了の手続き，RESET__Sというところでは星が画面から消え去ったあと，もう1回下から登場させるための手続きを行っているのです。

*　　*　　*

このプログラム，12月号のMMLとも仲がいいので，今月掲載されているソーサリアンのオープニングのテーマを流しながら星を斜めに流す，なんてことも可。ただし，CLEAR &HFBB0は忘れずに先頭につけてください。

リスト1 STARS

```
FBB0 3E CD 32 74 06 21 E5 FB : B8
FBB8 22 75 06 CD F2 FB 21 33 : AB
FBC0 FD 22 1F FD 21 84 FC 22 : FE
FBC8 5A 00 F3 ED 4B E1 FB 3E : 9F
FBD0 58 ED 79 ED 4B E3 FB 3E : 12
FBD8 A7 ED 79 3E 48 ED 79 FB : F4
FBE0 C9 A0 1F A1 1F FE 09 30 : 7F
FBE8 05 3E 20 ED 79 AF CB A0 : E3
FBF0 03 C9 21 33 FD 06 4A C5 : 32
FBF8 CD FF FB C1 10 F9 C9 36 : 90
FC00 08 E5 CD 58 FC 7C 3C E6 : AC
FC08 07 20 02 3E 05 F6 20 E1 : 63
FC10 23 77 E5 CD 58 FC CB 3C : A7
FC18 7C FE 67 D2 13 FC FE 17 : D7
FC20 DA 2F FC D6 17 26 00 6F : 87
FC28 01 30 37 09 C3 3F FC 26 : 95
---------------------------------
SUM: DD BD E5 EC E2 CC 79 41 E545

FC30 00 6F 29 29 29 29 54 5D : C4
FC38 29 29 19 01 00 30 09 EB : 90
FC40 E1 23 73 23 72 E5 CD 58 : 16
FC48 FC 7C 3C E6 07 20 02 3E : 01
FC50 01 E1 23 77 23 77 23 C9 : 02
FC58 ED 5B 6C FC ED 4B 6F FC : 53
FC60 CD 71 FC 22 6C FC ED 5F : 10
FC68 32 6E FC C9 23 E1 00 83 : EC
FC70 03 21 00 00 3E 10 29 CB : 66

FC78 23 CB 12 D2 7F FC 09 3D : 93
FC80 C2 76 FC C9 ED 73 31 FD : 8B
FC88 31 31 FD F5 C5 D5 E5 08 : DB
FC90 F5 3E 24 08 2A 1F FD 11 : B6
FC98 04 00 19 35 C2 EE FC 23 : 21
FCA0 7E 2B 77 2B 46 2B 4E CB : D5
FCA8 A0 ED 78 E6 20 28 09 AF : EB
---------------------------------
SUM: 23 3B AF 6F 02 B1 43 40 6852

FCB0 ED 79 CB E0 3E 20 ED 79 : D5
FCB8 CB E0 2B 2B 35 C2 E0 FC : D4
FCC0 EB 60 69 01 51 00 B7 ED : AA
FCC8 42 EB 7B D6 50 7A DE 30 : 56
FCD0 DA 16 FD 23 23 73 23 72 : 3B
FCD8 42 4B 3E 08 2A 1F FD 77 : 90
FCE0 ED 78 FE 21 30 08 04 ED : AD
FCE8 A3 CB A0 04 ED A3 2A 1F : EB
FCF0 FD 11 06 00 19 7D D6 EF : 6F
FCF8 7C DE FE DA 01 FD 21 33 : 84
FD00 FD 22 1F FD 08 3D C2 93 : D5
FD08 FC F1 08 E1 D1 C1 F1 ED : 46
FD10 7B 31 FD FB ED 4D 2A 1F : 27
FD18 FD CD FF FB C3 F5 FC 00 : 78
FD20 00 00 00 00 00 00 00 00 : 00
FD28 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 7B 48 DA E0 21 53 80 48 DA7A
```

表3 リスト1の変更点

CZ-8BM2 (RS-232Cマウスボード)をお持ちの方 POKE &HFBE1, &HA8, &H1F:POKE &HFBE3, &HA9, &H1F
CZ-8BR1(立体映像セット) をお待ちの方 POKE &HFBE1, 4, 10:POKE &HFBE3, 5, 10
CZ-8BS1(FM音源ボード)をお持ちの方 POKE &HFBE1, 4, 7:POKE &HFBE3, 5, 7

リスト2 PCG定義

```
10 DEFCHR$(32)=STRING$(24,0)
20 DEFCHR$(0)=STRING$(24,0)
30 DEFCHR$(1)=HEXCHR$("800000000000000080000000000000008000000000000000")
40 DEFCHR$(2)=HEXCHR$("004000000000000000400000000000000040000000000000")
50 DEFCHR$(3)=HEXCHR$("000020000000000000002000000000000000200000000000")
60 DEFCHR$(4)=HEXCHR$("000000100000000000000010000000000000001000000000")
70 DEFCHR$(5)=HEXCHR$("000000000800000000000000080000000000000008000000")
80 DEFCHR$(6)=HEXCHR$("000000000004000000000000000400000000000000040000")
90 DEFCHR$(7)=HEXCHR$("000000000000020000000000000002000000000000000200")
100 DEFCHR$(8)=HEXCHR$("000000000000000100000000000000010000000000000001")
```

リスト3 STARSソースリスト

```
0000              ;----------------
0000                ;   STARS
0000              ;----------------
0000                OFFSET          8000H-0FBB0H
FBB0                ORG             0FBB0H
0006 P            DATA: EQU         6         ;データの長さ
004A P            HOWM: EQU         74        ;星の長さ
0017 P            YMAX: EQU         23        ;Y軸のMAX
0001 P            YTOP: EQU         1         ;Y軸のMIN
FEEF P            BUFEND:           EQU       DATA*HOWM+STARVAL
00FE P            BUFENDH:          EQU       BUFEND/256
00EF P            BUFENDL:          EQU       -BUFENDH*256+BUFEND
FBB0
3050 P            TOPMIN:           EQU       YTOP*80+3000H
0030 P            TOPMINH:          EQU       TOPMIN/256
0050 P            TOPMINL:          EQU       -TOPMINH*256+TOPMIN
FBB0
FBB0              STAR:
FBB0 3E CD          LD  A,0CDH                ;CALL
FBB2 32 74 06       LD  (0674H),A
FBB5 21 E5 FB       LD  HL,ROLL_PATCH1
FBB8 22 75 06       LD  (0675H),HL
FBBB
FBBB CD F2 FB       CALL  TOROKU
FBBE 21 33 FD       LD  HL,STARVAL
FBC1 22 1F FD       LD  (KIX),HL
FBC4
FBC4 21 84 FC       LD  HL,INT_STAR           ;割り込みが実際におこった場合に
は、
FBC7                                          ;このルーチンにとんでいく。
FBC7 22 5A 00       LD  (005AH),HL            ;0058H+0002H=005AH
FBCA F3             DI
FBCB ED 4B E1 FB    LD  BC,(CTC0)
FBCF 3E 58          LD  A,58H
FBD1 ED 79          OUT (C),A                 ;割り込みベクトル設定
FBD3
FBD3 ED 4B E3 FB    LD  BC,(CTC1)
FBD7 3E A7          LD  A,167
FBD9 ED 79          OUT (C),A
FBDB 3E 48          LD  A,72
FBDD ED 79          OUT (C),A
FBDF FB             EI
FBE0 C9             RET
FBE1
FBE1 A0 1F        CTC0: DW          1FA0H     ;ノーマルX1の人はここを704H (FM
音源装着時)
FBE3 A1 1F        CTC1: DW          1FA1H     ;ノーマルX1の人はここを705H (FM
音源装着時)
FBE5                                          ;また、FM音源をもっていない人で
、次のボードを
FBE5                                          ;お持ちの方は
FBE5                                          ;CZ-8BR1 : CTC0 = A04H , CTC1 =
A05H
FBE5                                          ;CZ-8BM2 : CTC0 =1FA8H , CTC1 =1
FA9H
FBE5                                          ;に変更して下さいね。X1turboの人
はこのままでOK
FBE5              ROLL_PATCH1:
FBE5 FE 09          CP  9
FBE7 30 05          JR  NC,RP01
FBE9 3E 20          LD  A," "     ;IF A=<8
FBEB ED 79          OUT (C),A
FBED AF             XOR A
FBEE              RP01:
FBEE CB A0          RES 4,B
FBF0 03             INC BC
FBF1 C9             RET
FBF2
FBF2              TOROKU: ;             0     1       2       3
   4      5
FBF2                ;ﾃﾞｰﾀ形式       STAR DATA/COLOR/GRAM ADR low/GRAM ADR hi
gh/SPEED/SPEED'
FBF2
FBF2 21 33 FD       LD  HL,STARVAL
FBF5 06 4A          LD  B,HOWM
FBF7              HOWM_LP01:
FBF7 C5             PUSH  BC
FBF8 CD FF FB       CALL  SETVAL
FBFB C1             POP BC
FBFC 10 F9          DJNZ  HOWM_LP01
FBFE C9             RET
FBFF
FBFF              SETVAL:
FBFF 36 08          LD  (HL),8    ;0
FC01 E5             PUSH  HL
FC02              IRO:
FC02 CD 58 FC       CALL  RND
FC05 7C             LD  A,H
FC06 3C             INC A
FC07 E6 07          AND 7
FC09 20 02          JR  NZ,IRO0
FC0B 3E 05          LD A,5
FC0D              IRO0:
FC0D F6 20          OR  32
FC0F E1             POP HL
FC10 23             INC HL
FC11 77             LD  (HL),A    ;1
FC12 E5             PUSH  HL
FC13              SAIRN:
FC13 CD 58 FC       CALL  RND
FC16 CB 3C          SRL H
FC18 7C             LD  A,H
FC19 FE 67          CP  80+YMAX
FC1B D2 13 FC       JP  NC,SAIRN
FC1E FE 17          CP  YMAX
FC20 DA 2F FC       JP  C,MIGIKARA
FC23 D6 17            SUB YMAX
FC25 26 00            LD  H,0
FC27 6F               LD  L,A
FC28 01 30 37         LD  BC,YMAX*80+3000H
FC2B 09               ADD HL,BC
FC2C C3 3F FC         JP  FI
FC2F              MIGIKARA:
FC2F 26 00            LD  H,0
FC31 6F               LD  L,A
FC32 29               ADD HL,HL   ;*2
FC33 29               ADD HL,HL   ;*4
FC34 29               ADD HL,HL   ;*8
FC35 29               ADD HL,HL   ;*16
FC36 54               LD  D,H
FC37 5D               LD  E,L
FC38 29               ADD HL,HL   ;*32
FC39 29               ADD HL,HL   ;*64
FC3A 19               ADD HL,DE   ;80
FC3B 01 00 30         LD  BC,3000H
FC3E 09               ADD HL,BC
FC3F              FI:
FC3F EB             EX  DE,HL
FC40 E1             POP HL
FC41 23             INC HL
FC42 73               LD  (HL),E          ;2
FC43 23             INC HL
FC44 72               LD  (HL),D          ;3
FC45 E5             PUSH  HL
FC46              SPEDSET:
FC46 CD 58 FC       CALL  RND
FC49 7C               LD  A,H
FC4A 3C               INC A
FC4B E6 07            AND 7
FC4D 20 02            JR  NZ,SONOMAMA
FC4F 3E 01            LD A,1
FC51              SONOMAMA:
FC51 E1             POP HL
FC52 23             INC HL
FC53 77               LD  (HL),A          ;SPEED
FC54 23             INC HL
FC55 77               LD  (HL),A          ;SPEED'
FC56 23             INC HL
FC57 C9               RET
FC58
FC58              RND:
FC58 ED 5B 6C FC    LD  DE,(OLDRND)
FC5C ED 4B 6F FC    LD  BC,(STEP)
FC60 CD 71 FC       CALL  MULTI
FC63 22 6C FC       LD  (OLDRND),HL
FC66 ED 5F          LD  A,R
FC68 32 6E FC       LD  (REFR),A
FC6B C9             RET
FC6C
FC6C 23 E1        OLDRND: DW        0E123H
FC6E              REFR:   DS        1
FC6F 83 03        STEP:   DW        899
FC71
FC71              MULTI
FC71 21 00 00       LD  HL,0
FC74 3E 10          LD  A,10H
FC76              MLOOP:
FC76 29             ADD HL,HL
FC77 CB 23          SLA E
FC79 CB 12          RL  D
FC7B D2 7F FC       JP  NC,SKIP
FC7E 09             ADD HL,BC
FC7F              SKIP
FC7F 3D             DEC A
FC80 C2 76 FC       JP  NZ,MLOOP
FC83 C9             RET
FC84
FC84              ;------------------
FC84                ; ほし  わりこみ
FC84              ;------------------
FC84              INT_STAR:
FC84 ED 73 31 FD    LD  (INTSP),SP
FC88 31 31 FD       LD  SP,INTSP
FC8B F5             PUSH  AF
FC8C C5             PUSH  BC
FC8D D5             PUSH  DE
FC8E E5             PUSH  HL
FC8F 08             EX  AF,AF'
FC90 F5             PUSH  AF
FC91
FC91 3E 24          LD  A,36
FC93              HOSHI_LOOP:
FC93 08             EX  AF,AF'
FC94 2A 1F FD       LD  HL,(KIX)
FC97
FC97 11 04 00       LD  DE,4
FC9A 19             ADD HL,DE
FC9B
FC9B 35             DEC (HL)      ;+4
FC9C C2 EE FC       JP  NZ,SKIP_S
FC9F 23             INC HL
FCA0 7E             LD  A,(HL)    ;+5
FCA1 2B             DEC HL
FCA2 77             LD  (HL),A    ;+4
FCA3 2B             DEC HL
FCA4 46             LD  B,(HL)    ;+3
FCA5 2B             DEC HL
FCA6 4E             LD  C,(HL)    ;+2
FCA7
FCA7 CB A0          RES 4,B
FCA9 ED 78          IN  A,(C)
FCAB E6 20          AND 32
FCAD 28 09          JR  Z,S_WK
FCAF AF             XOR A
FCB0 ED 79          OUT (C),A
FCB2 CB E0          SET 4,B
FCB4 3E 20          LD  A," "
FCB6 ED 79          OUT (C),A
FCB8              S_WK:
FCB8 CB E0          SET 4,B
FCBA
FCBA 2B             DEC HL        ;+1
FCBB 2B             DEC HL        ;+0
FCBC 35             DEC (HL)
FCBD C2 E0 FC       JP  NZ,FI2
FCC0 EB               EX  DE,HL
FCC1 60               LD  H,B
FCC2 69               LD  L,C
FCC3 01 51 00         LD  BC,81
FCC6 B7               OR  A
FCC7 ED 42            SBC HL,BC
FCC9 EB               EX  DE,HL
FCCA 7B               LD  A,E
FCCB D6 50            SUB TOPMINL
FCCD 7A               LD  A,D
FCCE DE 30            SBC A,TOPMINH
FCD0 DA 16 FD         JP  C,RESET_S         ;IF DE<TOPMIN GOTO RESET_S
FCD3 23               INC HL
FCD4 23               INC HL
FCD5 73               LD  (HL),E  ;+2
FCD6 23               INC HL
FCD7 72               LD  (HL),D  ;+3
FCD8 42               LD  B,D
FCD9 4B               LD  C,E
FCDA 3E 08            LD  A,8
FCDC 2A 1F FD         LD  HL,(KIX)
FCDF 77               LD  (HL),A
FCE0              FI2:
FCE0 ED 78          IN  A,(C)
FCE2 FE 21          CP  " "+1
FCE4 30 08          JR  NC,SKIP_S
FCE6 04             INC B
FCE7 ED A3          OUTI
FCE9 CB A0          RES 4,B
FCEB 04             INC B
FCEC ED A3          OUTI
FCEE              SKIP_S:
FCEE 2A 1F FD       LD  HL,(KIX)
FCF1 11 06 00       LD  DE,DATA
FCF4 19             ADD HL,DE
FCF5              HANTEI:
FCF5 7D             LD  A,L
FCF6 D6 EF          SUB BUFENDL
FCF8 7C             LD  A,H
FCF9 DE FE          SBC A,BUFENDH
FCFB DA 01 FD       JP  C,SETVAL_FOR_RET
FCFE 21 33 FD       LD HL,STARVAL             ;IF HL>=BUFEND THEN HL=STARVAL
FD01              SETVAL_FOR_RET:
FD01 22 1F FD       LD  (KIX),HL
FD04 08             EX  AF,AF'
FD05 3D             DEC A
FD06 C2 93 FC       JP  NZ,HOSHI_LOOP
FD09
FD09 F1             POP AF
FD0A 08             EX  AF,AF'
FD0B E1             POP HL
FD0C D1             POP DE
FD0D C1             POP BC
FD0E F1             POP AF
FD0F ED 7B 31 FD    LD  SP,(INTSP)
FD13 FB             EI
FD14 ED 4D          RETI
FD16
FD16              RESET_S
FD16 2A 1F FD       LD  HL,(KIX)
FD19 CD FF FB       CALL  SETVAL
FD1C C3 F5 FC       JP  HANTEI
FD1F
FD1F 00 00        KIX:    DW          0
FD21                      DS        16
FD31              INTSP:  DS        2
FD33              STARVAL:          DS        DATA*HOWM
FEEF
```

アセンブラによるX68000料理教室

Nakamori Akira 中森　章
Kuwano Masahiko 桒野 雅彦

X68000には，質のいいアセンブラとエディタが用意されています。とはいっても，具体的にどのようにやればアセンブラを活用できるのかわからない人も多いことでしょう。このコーナーでは，X68000でアセンブリ言語を使ったプログラミングをするうえで必要となる知識を解説していきたいと思います。

X68000のプログラム開発環境（言語プロセッサ）としては，X-BASIC，アセンブラ，Cコンパイラがあります。私たちは，これらの言語プロセッサを使ってプログラムを作っていくわけですが，X68000のハードウェアの奥の奥までをいじるためには，やはりアセンブラに頼らなければなりません。また，プログラムの実行速度をひたすら速くしたいというときにもアセンブラは不可欠です。

最近ではオリジナルな開発ツールのほかにプリプロセッサ（PP68K）とか新アセンブラ（CMA68K）がシャープ以外からも発売されるようになりました。アセンブラを用いたアセンブリ言語によるプログラムの開発環境がさらに充実しつつあるのはユーザーとしてはうれしいことです。やはり，「パーソナル」なコンピュータであるからにはマシンの全機能が扱えなければ面白くありませんからね。

アセンブラの基礎知識

X68000でアセンブリ言語によるプログラミングを行うのにぜひとも必要なのがアセンブラとリンカです。アセンブラはアセンブリ言語で記述したプログラム（ソースプログラム）を機械語に翻訳する道具，リンカはアセンブラによって作られた機械語プログラム（オブジェクトプログラム）をつなぎ合わせて（リンクという）実行形式のプログラムを作る道具です[1]。

アセンブラでいったん機械語に翻訳するのに，なぜわざわざリンカを使う必要があるのかと疑問を持つ人がいるかもしれませんね。これは，ひと言でいえば分割アセンブルに対応するためです。

アセンブリ言語で書くソースプログラムは，1行に1命令しか記述できないことが多く，非常に長くなります。数百行程度のプログラムならまだしも，それが何千行，数万行という長さになってくるとデバッグが大変ですし，アセンブルにかかる時間も非常に長くなります。つまり，開発効率が低下します。また，数万行というソースプログラムをアセンブルできるアセンブラがあるかどうかも疑問です。そこで，プログラムを幾つかのモジュールに分割してアセンブルしておき，あとからそれらのオブジェクトプログラムをつなぎ合わせるという方法がとられます。これが分割アセンブルです。

こうすれば過去に作ったプログラムを再利用することも簡単になり，新規に作らなければならないプログラムの行数を少なくできます。また，ソースプログラムのなかにはモジュールに分割するほど規模の大きくないプログラムもありますが，多くの場合はキーボードから入力した文字を受け取ったり，画面に文字を表示するためになんらかのライブラリ[2]で供給されるオブジェクトプログラムとリンクして実行形式のプログラムを作るため，やはりリンクという作業は必要です。

ところで，本当にソースプログラムがそれ自身で完結していて，別プログラムで定義してある関数や変数を参照しない場合もあります（趣味で作るプログラムはこっちの場合が多い）。そういう場合はリンカをわざわざ使う理由はないと思うのですが，アセンブルから実行形式の処理手順を統一するためか，必ずリンカを使わなければならないのが実情です。今回の記事で例題として示すソースプログラムはすべて自分自身で完結しているものですが，それらのプログラムを実行するためには，

1) アセンブル
2) リンク

という2段の操作が必要です。

アセンブラとリンカの用意ができたら即プログラミングといきたいところですが，その前に必要最低限の基礎知識について復習しておきましょう。

1) 命令

X68000にはモトローラ社のMC68000というCPUが使われています。したがって，ソースプログラムとしてどのような命令が記述できるのかはMC68000のユーザーズマニュアルなどを参照して，MC68000にどのような命令があるのかを知る必要があります。アセンブラのマニュアルには命令の説明があることが多いのでユーザーズマニュアルを特に買う必要はないでしょう（けど，買ったほうがよい）。

2) 疑似命令

CPUが同じであれば，マシン語の命令セットが同じになるのは当たり前です。そして，アセンブラの使い勝手を特徴づけるのが疑似命令です。最近発売されたCMA68K（シティソフト）というアセンブラは，C言語と同じプリプロセッサ機能（要は疑似命令）を「売り」にしていましたね。

疑似命令はアセンブラに対する命令ですから，それぞれのアセンブラに固有のものです。しかし，違うアセンブラを使うごとに疑似命令を使い分けなければならない，かというとそうでもありません。すべてのアセンブラに標準的に必ず備えられている疑似命令があり，実際はそれを知っているだけで，なんとかやっていくことができます。それはEQUとENDという疑似命令です[3]。EQUはシンボルの名前を変更する疑似命令，ENDはソースプログラムの終わりを示す疑似命令です。

EQUは多くの場合，定数値に名前を持たせるために使用します。たとえば，

```
MAXNUM    EQU    256
```

という記述はソースプログラム中に出てきたMAXNUMという名は256という数値に置き換えてアセンブルされることを指示します。

一方，ENDはこれ以降の記述をアセンブルしないように指示します。省略される場

合が多いのですが，プログラムの実行アドレスを指示するために使用することができます。たとえば，

END　KAISHI

という記述は，プログラムがリンカで実行形式に変換されたときに，KAISHIというラベルで示すアドレスが実行開始アドレスになるように指示します。ENDで特に指示しない場合，プログラムの実行はプログラムの先頭から開始されます。

EQUとENDのほかに知っておかなければならない疑似命令に，値の初期化付きで領域を確保する疑似命令（DC.B/DC.W/DC.L），および領域のみを確保する疑似命令（DS.B/DS.W/DS.L）がありますが，これはアセンブラごとに若干記述の違いがあるようです。これらについては各自勉強してください。

3）マクロ

アセンブリ言語によるプログラムは行数が長くなる傾向にあります。そこで，よく現れる命令列をひとまとめして名前を付けてやるとプログラムの行数を少なくすることができ，また，プログラムを読みやすくすることができます。マクロとはそのための機能です。

マクロの記述には引数やローカルなラベルを使えるので，マクロをちょっとしたサブルーチン代わりに利用することもできます。サブルーチンとマクロの違いは，サブルーチンはひとつだけ存在する命令列の実体をJSR命令やBSR命令で参照するのに対して，マクロはそれが記述されるごとにその場所に命令列を埋め込んでいくという点です[4]。マクロはどうしても必要なものではありませんが，使ってみると便利な機能でしょう。

MC68000用のアセンブラでよく見られるマクロ定義として，

```
PUSH    MACRO   REGLIST
        MOVEM.L REGLIST,-(SP)
        ENDM
```

があります。PUSHがマクロの名前，REGLISTがマクロへの引数，マクロの実体がMCAROとENDMの間にある，

```
MOVEM.L   REGLIST,-(SP)
```

です。このマクロが定義されている場合，ソースプログラム中に，

```
PUSH   D0-D1
```

などという記述があると，そこは，

```
MOVEM.L   D0-D1,-(SP)
```

に置き換えられてアセンブルされます。この場合，プログラムの行数は短くなりませんが見やすさが向上します。

アセンブルの手順

ソースプログラムができたら，アセンブラを使ってアセンブルをしましょう。仮にファイル名がTEST.Sだとしましょう。このように，アセンブリ言語で書いたプログラムの入ったファイルの名前の拡張子は習慣的にS（Sourceの略）にします。

また，ここではX68000の標準的なアセンブラAS.Xを使う場合にそって説明します。ほかのアセンブラでも，操作は似たり寄ったりなので心配ありません。

TEST.Sをアセンブルするためには，Human68kのCOMMAND.Xのプロンプトが出ているところで，

AS　TEST.S

と打ち込みます。ただし，アセンブルが実行できるためにはカレントディレクトリにAS.Xがあるか，環境変数PATHで指定されたディレクトリにAS.Xがなければなりません。アセンブルが正常に終了すると，

No Fatal error(s)(致命的エラーナシ)

とメッセージが出て，ファイル名の拡張子のSをO（Objectの略）に置き換えたTEST.Oというオブジェクトファイルがソースプログラムと同じディレクトリにできます。

もし，アセンブラがソースプログラムをどのような機械語に翻訳したかを見たければ，アセンブル時に，

AS　/p　TEST.S

と/pスイッチを付けて打ち込みます。これでアセンブルした結果であるアセンブルリストが，ソースプログラムの拡張子をSからPRN（Printの略）に置き換えたTEST.PRNというファイル名で作られます。

アセンブルの次はリンクでしたね。リンクはアセンブルしたときと同じ要領で，

LK　TEST.O

と打ち込みます。カレントディレクトリにLK.Xがあるか，環境変数PATHで指定したディレクトリにLK.Xがなければならないのはアセンブル時と同じです。何もメッセージが出なければリンクは正しく終了しました。オブジェクトプログラムであるTEST.Oと同じディレクトリに実行形式のTEST.Xというファイルが作られます。拡張子はOからX（Executableの略）に変わります。

さて，プログラムを実行するときには，実行形式のファイル名を，

TEST

と打ち込みます。このとき.Xは省略して構いません。するとプログラムの実行が始まります。

以上でアセンブラ自体の基礎知識はおしまいです。次に，実際のプログラミングのために役立つDOSコールやIOCSコールについて触れておきましょう。

DOSコールを使う

アセンブリ言語でプログラミングをする場合，第一の障害はパソコンとの対話部分をどうするかです。何かの関数から呼ばれるサブルーチンなどを作る場合ならいざしらず，それ自身で完結したプログラムを作る場合には，データをキーボードから入力したり，得られた結果をディスプレイに表示したりするといったパソコンとの対話が必要になります。

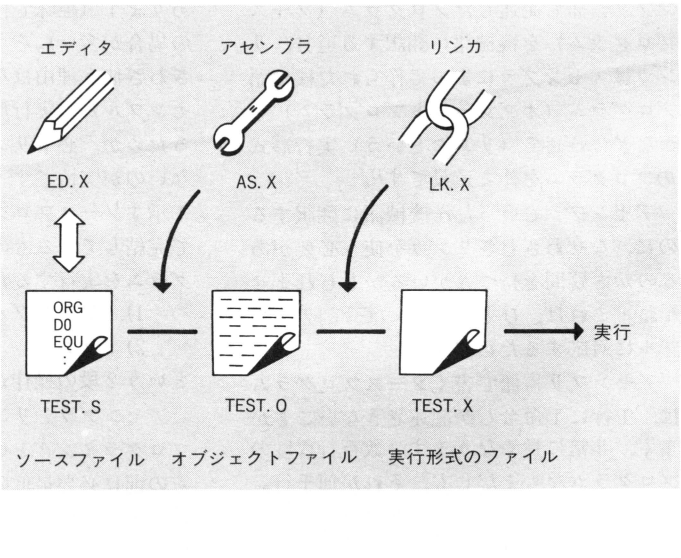
入出力を制御するためにはパソコンのハードウェアに直接触らなければならず，初心者には手が出ません。そこで利用したいのが，OSに用意されているファンクションコール[5]，つまりDOSコールと呼ばれるものです。これは，OSにあらかじめ用意されているサブルーチン群をコールすることによって入出力などの複雑な動作を行わせる機能です。

X68000のHuman68kの場合は未定義命令，

$0FF\times\times_H$

（××はファンクション番号）

図1　アセンブルの手順



を実行することでDOSコールを行います。このDOSコールを使えば誰でも手軽にアセンブリ言語によるプログラムが書けるようになるのです。また，DOSコールのなかには「プログラムの実行を終了させる」という，アセンブリ言語でのプログラミングに不可欠な機能がありますからチェックしておきましょう。

DOSコールを実際に行うためにはスタックに必要なだけの引数を積み，0FF××$_{H}$命令を実行します。そしてDOSコールからリターンしたあと，スタックポインタを補正してスタックに積んだ引数を捨てます。

さて，実際にプログラムの目的が決まったら，使えそうなDOSコールを探します。DOSコールの種類を知るためにはCやTHE福袋V2.0のプログラマーズマニュアルを見ればいいでしょう。

IOCSコールを使う

DOSコールはOS（Human68k）と一体となったサブルーチン群のコールです。しかし，X68000にはROMの中にDOSコールよりももっと基本的な入出力を行うためのサブルーチン群があります。これが IOCS（Input Output Control System）と呼ばれるもので，これらのサブルーチン群の機能を使うためにIOCSコール[6]が用意されています。

ROMとして標準装備されているサブルーチン群ですからOSが変わっても変わることはありません。言い換えればIOCSコールはどんなOS上であっても共通に使える機能なのです。当然，周辺機器とデータをやり取りするための基本的な機能はすべてROMの中にあると考えてよいでしょう。

IOCSコールはファンクション番号（0～255）をレジスタD0に設定し，MC68000のTRAP命令の15番を実行することで実行されます。つまり，

```
MOVEQ.L #ファンクション番号,D0
TRAP    #15
```

という命令列で実行されます。また，もし引数が必要なときはD0以外のレジスタで渡すようになっています。

I/Oポートとワークエリアを使う

これまで説明してきたDOSコールとIOCSコールをうまく組み合わせて使えばX68000のほとんどすべての機能を使うことができます。DOSコールはOSに，IOCSコールはROMに対してファンクション番号を指定してサービス要求を出し，その処理結果を受け取る仕組みになっています。

まあ，それがOSというものなのですが，言ってみればこれは間にご用聞きを立てた仕事の発注みたいなもので，システムの中でいったい何が起こっているのかは知る由もありません。あるファンクションコールなりIOCSコールの処理速度が非常に遅いとしても，コールをする側はそれが終わるのを指をくわえて見ているしかありません。

DOSコールやIOCSコールは，OSやシステムを設計した人が考えていた使い方（たとえば，CRTモードが19種類設定されているということなど）の範囲では非常に有効ですが，それ以上のわがままを要求したいとなると話は別です。たとえば，CRTモードを操作する場合を例にとってみればわかるように，テキスト画面だけの設定だけしか必要ないときも，必ずグラフィック画面とスプライト画面の設定を行うという余計な処理が入ってしまいます。もっと高速なプログラムを望む人にとってはなんとかしたくなるでしょう。

そんなとき，ファンクションコールやIOCSコールを通さず，ハードウェアに対して直接指示を出したいと思うことがあるはずです。そういう理由から，CPUの周辺のハードウェア（IC）とのインタフェイスであるI/OポートやOSの使用するワークエリアを直接いじりたいという欲求が生まれるのは当然のことでしょう。X68000のI/Oポートに関しては小学館の『X68000データブック』などで公開されていますから，少し勉強すればパソコンのハードウェアを自由自在に扱うことも可能なのです。

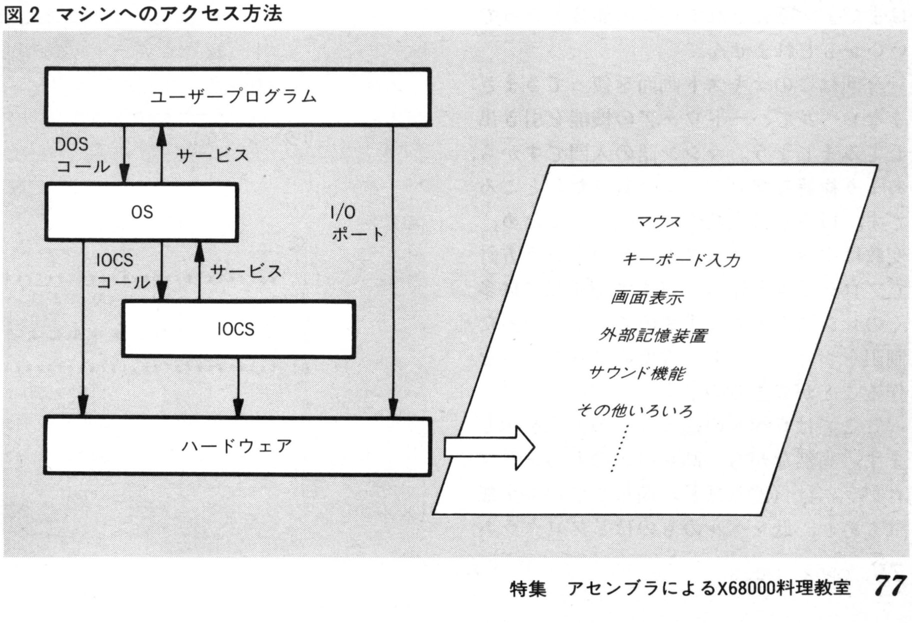
とはいえ，メーカーが直接I/Oポートを公開しているのではないということが重要。直接I/Oポートを操作したことによってパソコンが壊れたとしても誰も責任を取ってくれません。パソコンがそう簡単に壊れるとは思えませんが，そうなっても笑っていられるだけの覚悟がある人ならI/Oポートを直接いじる価値があるでしょう。そこまでやるからには，何が起きても個人の責任ということです。

＊　＊　＊

さあ，これで予備知識の整理と心の準備はできたと思います。いよいよ次のページからはアセンブラを使った実際のプログラミングに挑戦してみましょう。

1) アセンブラ，リンカ自体も実行形式のプログラムのひとつ。最近ではC言語で記述されることが多いが，アセンブルやリンクの時間を高速にしたい場合はアセンブリ言語で記述する。なかなかプログラムのアセンブルが終わらない場合，「このアセンブラ，Cで書かれてんじゃないの」などと悪口を言う。
2) ライブラリとは，オブジェクトプログラムを集めてひとつのファイルにしたもの。リンカは指定されたライブラリから必要なオブジェクトプログラムのみを取り出して，ほかに指定されたオブジェクトプログラムと結合し，実行形式のプログラムを作成する。ライブラリを作成するための道具をアーカイバ（ライブラリアンともいう）という。
3) 疑似命令の名前は「．」（ピリオド）で始まる場合が多い。しかし，現実には「．」を省略した名前も使えることがある。
4) マクロは「開いたサブルーチン」と呼ばれることがあるそうだ。もちろん，通常のサブルーチンは「閉じたサブルーチン」である。
5) C compiler PRO-68KのシステムディスクのincludeというディレクトリにDOSCALL.MACというファイルがあり，このファイルの中にすべてのDOSコールの定義がある（EQU疑似命令による定義が並んでいる）。
6) C compiler PRO-68KのシステムディスクのincludeというディレクトリにIOCSCALL.MACというファイルがあり，このファイルの中にすべてのIOCSコールの定義がある（EQU疑似命令による定義が並んでいる）。

図2　マシンへのアクセス方法



初めは誰でも文字表示

まずはマシン語入門の基本である文字表示から。基本は基本でもX68000ならいくらでも発展していくテーマです。簡単なプログラムですから実際にアセンブルして結果を確認してみてください。

Kuwano Masahiko 桒野 雅彦

アセンブラによるX68000料理教室

古来からマシン語入門というのは，まず画面に文字を表示することから始まります。初めからマシン語のわかる人なんているはずがありません。なんの苦労もせずにマシン語を修得できるはずもなく，地道な努力を続けることがマシン語修得への唯一の道ともいえます。その最初のステップともいえるのが，文字表示です。どんなマシン語の達人でも，最初は命令表と首っ引きでプログラムを書き，ようやく画面に表示された1文字のAを見て感慨にふけったという経験があるものです。たとえX68000だろうが，マシン語入門の基本精神は変わりません。ただ，表示されるのが24×24ドットの高品位文字だったりするところが，16ビットマシンの違いというところでしょうか。

なぜ文字表示かというと，世の中の大半のマシンでは文字表示というのがもっとも簡単でわかりやすい例題だからです。X68000の場合，グラフィックを表示するのも，スプライトを動かすのも，FM音源を鳴らすのもそれほど手間は違いません。むしろ文字表示のほうがややこしいくらいのものです。これはX68000のテキスト画面がビットマップ/マルチフォントであることに起因しますが，いまのところビットマップの恩恵を被っている人はあまりいないでしょう。本来は非常に強力な武器になるはずのテキスト画面なのですが，現在のX68000ではまだまだ開発されていない部分といっていいかもしれません。

今回はこのテキスト画面を使ってさまざまなレベルでハードウェアの機能を引き出してみましょう。マシン語の入門ですから，あまり複雑なプログラムは避けたいところです。以下では変数管理を楽にするため，変数はすべてレジスタ上に持つという方針でプログラムを作っています。68000には多くのレジスタがありますので，こういった制限をつけても，ある程度のプログラムを作ることができるのです。

ここではレベルの違う3つの方法を示します。当然ながら，高レベルのものほどプログラムは（驚くほど）簡単ですが多少無駄もあり，低レベルのものほどプログラムは複雑で柔軟，高速な処理ができます。やっていることは「とにかく画面に文字を出す」と，ほとんど変わりません。

基本的な部分さえ理解すれば，あとはグラフィックだろうがスプライトだろうが，同じようなものです。これを基にマシン語への第1歩を踏み出してみてください。

DOSコールによる文字表示

小手調べとして，まずDOSコールを使ってみましょう。DOSコールの一覧を探すとDOSコール番号$FF02にputcharというものがあります。このDOSコールはスタックに文字コードを積んでから，

DC.W　$FF02

とすればその文字コードに相当する文字を表示するもので，使い方は実に簡単です。

リスト1を見てください。まず最初の.textは「以下は命令部分である」という意味の，次の.evenは「偶数番地から始める」という意味の宣言です。68000の命令コードのバイト数はすべて2の倍数となっており，命令は必ず偶数番地から始まるということになっています。もし奇数番地から始まるようであると直ちにアドレスエラーとなって，プログラムの実行は中断されてしまいます。このため，わざわざ.evenをつけて偶数番地からであるということをアセンブラに伝えてやっているわけです。いまのところは「おまじない」くらいに考えておいてもかまいません。

スタックの処理

次の行からが本題のプログラムです。@の文字コードをスタックに積み上げてDOSコールを実行します。ここで与える文字コードは本体についてきたBASICマニュアルにあるキャラクタコード表のとおりです。スタックに積むということは，スタックポインタと呼ばれるレジスタの値を減らして，

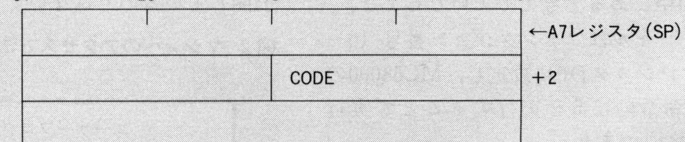
表1 putcharの機能

●putchar(CODE)　DOSコール番号：$FF02

機　能：文字コードにより文字を表示します

引　数：CODE(r)　word　1文字コード

コール：MOVE.W　CODE,−(SP)
　　　　DC.W　_PUTCHAR
　　　　ADDQ.L　#2, SP



リターン：なし

リスト1

```
*********************************************
*                   リスト1                  *
*                                           *
*         DOSコールによる文字表示             *
*                                           *
*********************************************
                .text                     * 命令領域である
                .even                     * 偶数番地から始めなさい
                move.w  #$40,-(sp)        * @の文字コードを積んで
                dc.w    $ff02             * DOSコール(putchar)
                addq.l  #2,sp             * 使ったスタックは元に戻す
                dc.w    $ff00             * 終了のDOSコール
                .end
```

その指す先にデータを転送する（CPUによってはあらかじめ転送してからレジスタの値を減らすものもある）という動作です。Z80や8086といったCPUではPUSHという専用の命令が用意されていますが，68000ではより汎用性を持たせて，MOVE命令に併合しています。「プリデクリメントアドレスレジスタ間接アドレッシング」というデラックスなアドレッシングモードをMOVE動作に適用すればすなわちPUSHであるというわけです。C言語風に書くなら，

　　＊(−−SP)＝0x40；

のようになります。

ちなみにこれと逆の動作はZ80などでは，やはりPOPという専用の命令で行いますが，68000では「ポストインクリメントアドレスレジスタ間接アドレッシング」をMOVE命令に適用して行います。

68000はこのようにスタック操作をMOVE命令に併合しており，スタックポインタ(SP）という特別な名前を持ったレジスタはありませんが，サブルーチンの呼び出しのときなどには，暗黙のうちにA7レジスタを使うことになっています。ほかのプロセッサと呼び方を合わせるならA7がスタックポインタということになりますし，一般的にA7と呼ぶよりもスタックポインタと呼んでおくほうがとおりがよいので，アセンブラでもA7の代わりにSPと書くやり方も許されています。ここではSPを使いましたが，A7と書いてももちろんかまいません。

さて，呼び出しが終わったら，スタックを元に戻しておきます。ワード（2バイト）のデータを積んでいましたので，スタックポインタを2だけ増やしておけばよいわけです。スタックポインタ（A7）に2を足せばよいわけで，ふつうのADD命令でもかまいませんが，ここではADDQ（クイックイミディエイト加算命令）を使ってみました。Z80や8086では1の加算や減算のためにINC/DEC（インクリメント/デクリメント）という専用の命令を用意したのですが，68000ではこれを拡張して1から8までカバーできる，クイックイミディエイト加算/減算命令を持たせており，実行速度も通常のADDよりも速くなっています。

最後にプロセス終了のDOSコールである，DOSコール番号$FF00（EXIT）を実行して終わりにします。サブルーチンから帰るときにはRTS命令を使いますが，プログラム自体を終了させるにはこれを使います。

そしてソースの最後には.endと書いて，アセンブラにソースの最後であることを教えてあげましょう。ファイルの最後にくれば終わりだということは自明のことなので すから，これはあったからといってどうということはないのですがないと寂しい，「へそ」のようなものです。礼儀としてつけるようにしておきましょう。

これをエディタから打ち込み，

```
AS      TEST1
LK      TEST1
```

として，アセンブル，リンクを行えば実行可能なTEST1.Xができます。改行をしていないので，@が表示されたすぐ横にCOMMAND.Xのプロンプトが出てくるはずです。

@だけではちょっと面白くないので，少しまとまった数の文字を出してみます（リスト2）。文字コードが40_Hから$7F_H$までの文字を連続して出力しましょう。ループを作るにはDBRA命令が便利です。これは指定されたレジスタの値を1だけ減らして，結果が−1でなければジャンプする命令ですから，ループしたい回数から1を引いた数をセットしておいて，ループの最後にDBRAをおいてループの先頭に戻ればよいわけです。

IOCSコールによる文字表示

1段階下の層に降りてIOCSコールを使ってみましょう。DOSコールと同じでは面白くないので，漢字コードを与えるとROMからの文字パターンを読み出して，指定したアドレスからそのデータを入れてくれるFNTGET（＄19）と文字パターンの画面への書き込みを行うTEXTPUT（＄1B）で行ってみましょう。その代わり，文字フォントは縦16ドットと24ドットの2種類を選べますし，文字を出す位置は1ドット単位で任意の場所を選べますから，「2ドットのスペース」といった，DOSコールではできない細かな操作も可能になります。ここではDOSコールでは表示されなかった24×24ドットのフォントで文字を出してみました（リスト3）。

プログラム自体は一直線ですので内容は極めて単純です。プログラムの.textセクションの始まる前に.bssとなっているのは，初期化されないデータ領域（dsで領域確保する場所）であることを宣言したものです。ちなみに初期化する領域（dwでデータをばらまく場所）は.dataで宣言します。X68000ではプログラムはすべてRAM上にロードされて動きますから，.textにデータ領域をとっても動かないことはありませんが，アセンブラ側でこのようなセクション区分を用意している以上，つきあってあげるのがエチケットというものでしょう。

次の.evenはやはり，以下は偶数番地から始めなさいという指示です。68000は奇数番地からワードアクセスしたりすると例によってアドレスエラーが出てしまうので，データはなるべく偶数番地から配置する指示をしたほうがよいようです。

リスト2

```
******************************************
*               リスト2                  *
*                                        *
*    DOSコールによる文字表示             *
*             (その2)                    *
*                                        *
******************************************
*               .include doscall.mac                  <----- これを使っても良い
_PUTCHAR        equ     $ff02
_EXIT                   equ     $ff00

                .text
                .even
character_output:                        * character_output()
                                         * {
                                         *       short   outdata,counter;
                move.w  #$40,d2          *       outdata = 0x40;
                move.w  #$3f,d1          *       counter = 0x40-1;
    character_out_loop:                  *       do {
                move.w  d2,-(sp)         *               putchar(outdata);
                dc.w    _PUTCHAR
                adda.l  #2,sp

                addq.w  #1,d2                    *               outdata++;
                dbra    d1,character_out_loop    *   } while (coutner--);

                move.w  #$0d,-(sp)               *       putchar(0xd);
                dc.w    _PUTCHAR
                adda.l  #2,sp

                move.w  #$0a,-(sp)               *       putchar(0xa);
                dc.w    _PUTCHAR
                adda.l  #2,sp

                dc.w    _EXIT                    * }
                .end
```

文字を並べる

ここまでは単純に上から順に流れていくだけげでした。さて，ここで縦横に文字を並べることを考えてみます。横に９文字を40ドットおきに，縦に10文字を30ドットおきに文字コード順に並べることにしましょう。

このためにはX，Yのそれぞれの方向について表示位置と表示した文字の数を勘定しておく変数が必要です。いまのプログラムに少し手を加えて二重ループで挟めばよいのですが，実際にプログラムを修正し始めるとそろそろ面倒な感じがしてくるのではないでしょうか。今回は「レジスタ以外に変数をとらない」という制約をつけましたから，だらだらと長いプログラムは自然と書けなくなっているはずです。書きにくくなってきたら，それが長さの限界ということですから，サブルーチンにして切り離すことを考えましょう。

サブルーチンにするのは簡単で，切り取ったプログラムの入り口にラベルをつけて，最後をRTS命令にするだけです。呼び出す側はBSR〈ラベル名〉とすればそれでOKです。ここでは画面の初期化と１文字表示をサブルーチンにしてみました（リスト４）。１文字表示の最初ではMOVEM命令でサブルーチン内で使う，あるいは中身を書き換える可能性のあるレジスタの値をスタックに待避しています。ここで待避したレジスタの値はRTSの寸前で元に戻しています。こうやって退避したレジスタはサブルーチンの中でいくら書き換えても帰る前に元に戻されますから自由に使うことができるわけです。これを上の二重ループの中から，呼び出すようにしています。

ハードウェア直接アクセスによる文字表示

いよいよ，いちばん下のレベルであるハードウェアの直接アクセスによる文字表示を行ってみましょう。ハードウェアといっ

リスト３

```
*******************************************
*            リスト３                     *
*                                         *
*    ＩＯＣＳコールによる文字表示         *
*                                         *
*******************************************
_EXIT                   equ     $ff00

iocs                    equ     $0f
_CRTMOD                 equ     $10             * 画面モードの設定
_TCOLOR                 equ     $15             * TEXTPUTで使うテキストVRAMの選択をする
_FNTGET                 equ     $19
_TEXTPUT        equ     $1b

                        .bss                    * 初期化されないデータ領域
                        .even                   * 偶数番地から始めます
        font_buffer:
                        ds.b    4+24*24/8       * 文字フォント格納領域

                        .text
                        .even

                        moveq.l #_CRTMOD,d0     * ７６８×５１２ドット・モード
                        move.w  #16,d1
                        trap    #iocs

                        moveq.l #_TCOLOR,d0     * プレーン１を使う
                        moveq.l #1,d1
                        trap    #iocs

                        moveq.l #_FNTGET,d0     * 文字パターンを読み出して
                        move.l  #$000c3026,d1
                        lea     font_buffer,a1
                        trap    #iocs

                        moveq.l #_TEXTPUT,d0    * 画面に表示する
                        move.w  #300,d1
                        move.w  #200,d2
                        lea     font_buffer,a1
                        trap    #iocs

                        dc.w    _EXIT
                        .end
```

表２ 今回使ったIOCSコール

$10 CRTMOD

CRT モードを指定。テキストをクリアして表示モードにしテキストパレットは標準に。グラフィック，スプライト，BGはクリアせず無表示モード

in	d1.w＝0	high	512×512	16/16	1024×1024
	＝1	low	512×512	16/16	1024×1024
	＝2	high	256×256	16/16	1024×1024
	＝3	low	256×256	16/16	1024×1024
	＝4	high	512×512	16/16	512×512
	＝5	low	512×512	16/16	512×512
	＝6	high	256×256	16/16	512×512
	＝7	low	256×256	16/16	512×512
	＝8	high	512×512	16/256	512×512
	＝9	low	512×512	16/256	512×512
	＝10	high	256×256	16/256	512×512
	＝11	low	256×256	16/256	512×512
	＝12	high	512×512	16/65536	512×512
	＝13	low	512×512	16/65536	512×512
	＝14	high	256×256	16/65536	512×512
	＝15	low	256×256	16/65536	512×512
	＝16	high	768×512	16/16	1024×1024
	＝−1	現在のモードを返します			
	＝$100＋上記	モード切り換えのみ行い画面クリア，パレット・コントラスト・表示モードの初期化はしない（データは順に解像度，表示エリア，テキスト/グラフィック色数，仮想画面の大きさを表す）			
out	d0.l＝	d1.wが−1のときは現在のモード，それ以外のときは壊れる			

$15 TCOLOR

テキスト VRAM を指定。IOCS コール$1a〜1cでアクセスされるテキスト VRAM を選択。コール終了後はテキスト VRAM を１に戻すこと

in	d1.b	b3|b2|b1|b0	
		0 |0 |0 |0	＝テキストプレーン1
		? |? |? |1	＝テキストプレーン1
		? |? |1 |0	＝テキストプレーン2
		? |1 |0 |0	＝テキストプレーン3
		1 |0 |0 |0	＝テキストプレーン4
		原則として1, 2, 4, 8の値のみとする	

$19 FNTGET

指定の漢字パターンを指定アドレスへ読み込む。スーパーバイザ領域指定可。バッファの大きさはフォントデータ＋４バイト必要

in	$0fに準ずる。ただしa1.lはデータバッファの先頭番地
out	0(a1).w＝文字パターンのＸ方向のドット数
	2(a1).w＝文字パターンのＹ方向のドット数
	4(a1).b＝パターンデータ

$1b TEXTPUT

指定ドット座標へパターンを書き込む。スーパーバイザ領域指定可

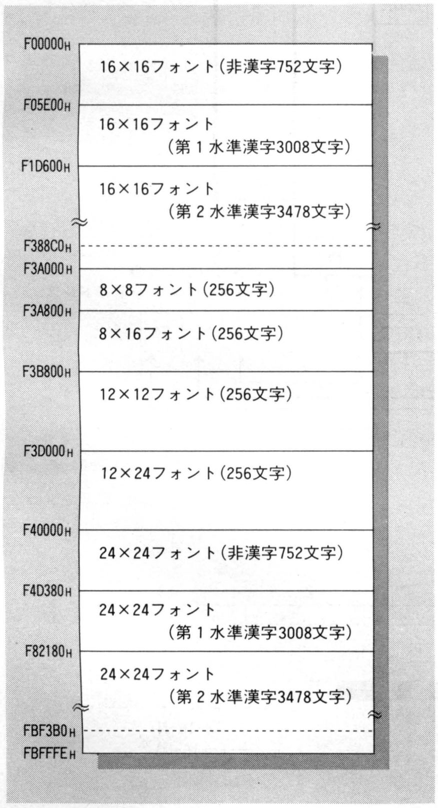
in	d1.w＝Ｘドット座標
	d2.w＝Ｙドット座標
	a1.l＝データバッファの先頭アドレス
	0(a1).w＝パターンのＸ方向のドット数
	2(a1).w＝パターンのＹ方向のドット数
	4(a1).b＝パターンデータ用のバッファ

ても，文字フォントを格納したCGROMとテキストVRAMだけが相手ですから，タイミングがどうしたという話にはなりませんが，多少のビット操作は入ってくることになります。

X68000が持っているCGROMは00F00000H番地からの768Kバイトの空間に割りつけられていて，この中に入っている文字フォントは24×24，16×16，12×24，12×12，8×16，8×8と，なんと6種類もあります。このうち漢字は24×24と16×16ドットの2種類で，ほかのものは英数字とかな文字専用になっています。BASICやIOCSでも，これらの一部しか使うことができないので，たとえば8×8ドットのフォントなどは完全に「死体」になっていますが，アセンブラでハードウェアのレベルまで手を出せば，これらを生かすことができます。

プログラムは最初，8×8ドットと16×16ドットだけにしようかと思っていたのですが，そのほかのものを作るのもたいした手間ではないので全フォントについて作ってみました。中を見てもわかるように，かなりよく似た処理を行っていますから，いくつかのパラメータを変更するだけでひとつ，ないしは2つのサブルーチンにまとめあげることも可能であろうと思います。これらのなかから，16×16ドットと8×8ドットの場合について説明しておきます。

図1 CGROMのアドレスマップ



の2つがわかってしまえば，ほかのフォントの表示ルーチンも容易に理解できるでしょう。

ハード構成を知る

まず，なにをおいても，まずCGROMとテキストVRAMの構造を知らないことには表示のやりようがありません。

CGROMのほうはパターンの種類こそ多いのですが，データの並びは比較的すんなりとできています。基本的に最初の1バイト目は文字の左上から横に8ドット分のデータで，左側が上位ビットになっています。横が8ドットより多いフォントでは，その右隣の8ビット，ないし4ビット（横12ドットのフォントの場合）が次のデータとなっており，右端までいったら次の列の左端

リスト4

```
***************************************
*              リスト4                *
*                                     *
*     IOCSコールによる文字表示        *
*            (その2)                  *
***************************************
_EXIT                   equ     $ff00

iocs                    equ     $0f
_CRTMOD                 equ     $10     * 画面モードの設定
_TCOLOR                 equ     $15     * TEXTPUTで使うテキストVRAMの選択をする
_FNTGET                 equ     $19
_TEXTPUT        equ     $1b

                .bss                    * 初期化されないデータ領域
                .even                   * 偶数番地から始めます
    font_buffer:
                ds.b    4+24*24/8       * 文字フォント格納領域

                .text
                .even
                                        * int   char_data,x,y;
                                        * short i,j;
character_output:                       * character_output()
                                        * {
                bsr     init_screen     *    init_screen();
                move.l  #$000c3021,d1   *    char_data = 0xc3021;
                moveq.l #100,d2         *    y = 100;
                moveq.l #9,d4           *    i = 9;
    char_out_loopy:                     *    do {
                move.l  #200,d3         *         x = 200;
                moveq.l #8,d5           *         j = 8;
    char_out_loopx:                     *         do {
                bsr     print_character *             print_charcter();
                addq.l  #1,d1           *             char_data++;
                add.l   #40,d3          *             x += 40;
                dbra    d5,char_out_loopx *         } while (j-- > 0);

                add.l   #30,d2          *         y += 30;
                dbra    d4,char_out_loopy *    } while (i-- > 0);

                dc.w    _EXIT           * }

*
*  一文字表示サブルーチン
*
*
* d1 = 表示文字のコードとサイズ
* d2 = 表示位置(Y座標)
* d3 = 表示位置(X座標)
*
print_character:
                movem.l d0-d3/a1,-(sp)  * レジスタ待避
                moveq.l #_FNTGET,d0     * 文字パターンを読み出して
                lea     font_buffer,a1
                trap    #iocs

                moveq.l #_TEXTPUT,d0    * 画面に表示する
                move.w  d3,d1
                lea     font_buffer,a1
                trap    #iocs

                movem.l (sp)+,d0-d3/a1  * レジスタ復帰
                rts

*
*  画面の初期化を行います
*
*
init_screen:
                moveq.l #_CRTMOD,d0     * 768×512ドット・モード
                move.w  #16,d1
                trap    #iocs

                moveq.l #_TCOLOR,d0     * プレーン1を使う
                moveq.l #1,d1
                trap    #iocs

                rts

                .end
```

から順に並んでいます。16×16ドットのフォントなら，1列目の左半分（8ドット），1列目の右半分，2列目の左半分……といって最後に16列目の右半分となるわけです。

一方，テキスト画面のほうはというと，これもよく似た構成になっていて，4つある各プレーンの左上から横方向に並んでおり左側が上位ビットになっています。テキスト画面は4プレーンあり，それぞれの左上隅のアドレスは00E00000$_H$，00E20000$_H$，00E40000$_H$，00E60000$_H$になっています。

テキスト画面は横1024×1024ドットの広さがあり，その一部がディスプレイに表示されているわけですから，1列下は1024/8＝128＝＄80を足したアドレスになります。文字を8ドットおき，すなわちバイト境界から出力するのであれば，この規則に従ってテキストVRAMとCGROMのアドレスだけを計算して，1バイトずつ読み出しては書き込むという作業を繰り返すだけでよいのですが，それではせっかくのビットマップ方式テキストVRAMが生かされません。やはりIOCSと同じように，1ドット単位で任意の場所に文字表示ができるように頑張ってみましょう。

表示位置をドット単位で

BASICのPSET的に攻めるなら，文字の左上のVRAM上でのアドレスとビット位置を計算しておいて，そこから順にフォントデータに従ってビットセット/リセットを繰り返す方法が思いつきます。確かに68000にはビットのセット/リセットを行う命令があり，対象とするビット位置はデータレジスタで指定できますからプログラムは案外簡単にすませられそうですが，24×24ドットのような大きなフォントを表示するときにはかなり時間がかかりそうな感じがします。

せめて1バイト分のデータはまとめて書き込むことを考えましょう。図4を見てください。いちばん上の列がテキストVRAMのイメージで，下の2つがCGROMから読み出したデータを書き込もうとしている位置であると思ってください。

上のように，ワード境界なら話は簡単で，CGROMからワードで読み出したデータをそのまま書き込めばよいわけです。厄介な感じがするのは下のように，中途半端な位置に書き込もうとしている場合です。横16ドットとはいっても図のように3バイトにまたがることになるわけで，ちょっと面倒そうです。しかし，ここで68000は内部構成が32ビットであったことを思い出せば，思いのほか簡単に片づくことに気がつくでしょう。

図の＋0から＋3までの4バイトをまとめて書き込むことを前提にして眺めれば，＋0のバイト境界のNドット目から16ドット分のデータを収めたいときは，そのデータを右端から16－Nビットだけ左にシフトしたロングワード（32ビット）データを作ればよいことがわかります。これを前提にして16×16ドットの文字出力を行っているputchar16x16を見てください。

図2　CGROMの内容

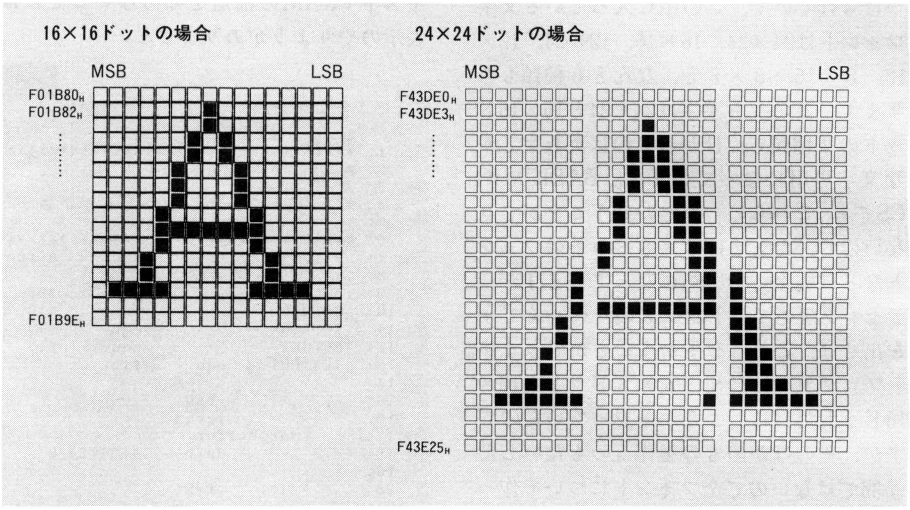


図3　テキスト実画面のアドレス配置

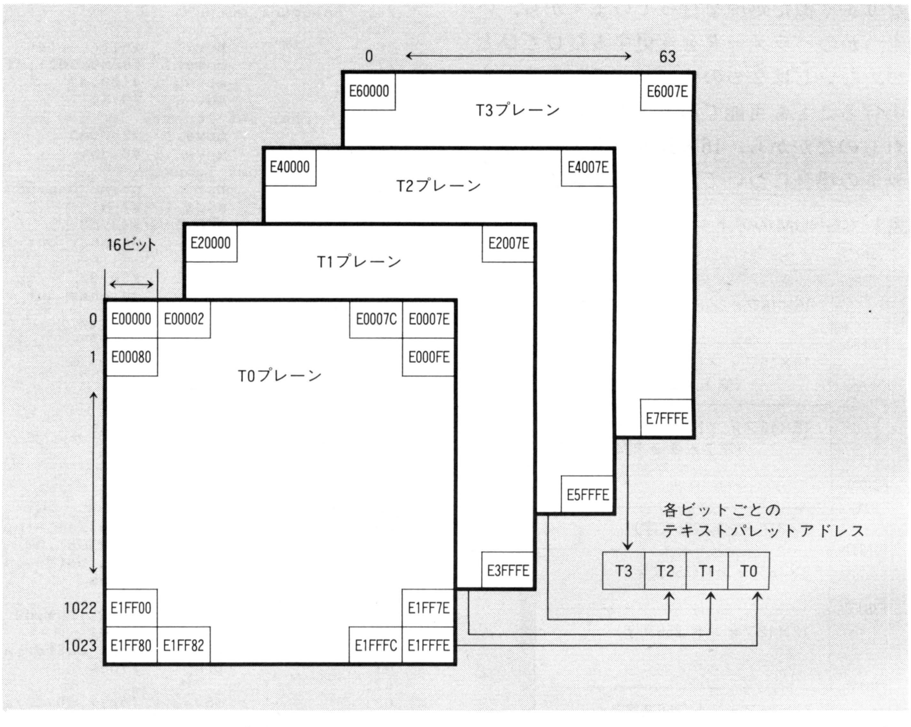
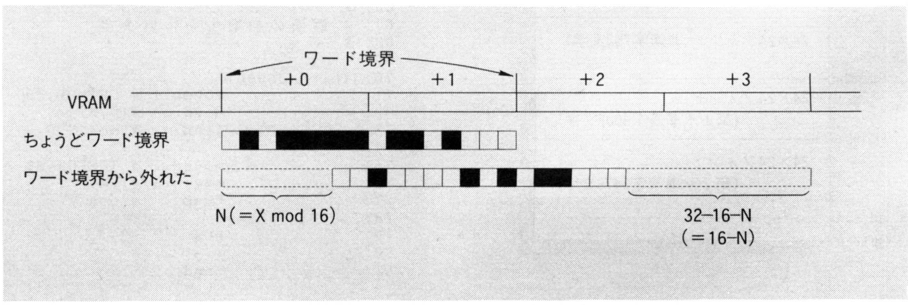


図4　16ドットフォントの書き込み



まず，例によってMOVEM命令でレジスタの内容を退避しておきます。次のADDで文字コードに＄9Bを足しているのは＄41が入ってきたときにAの文字が出るようにするためにやってみただけのことで，他意はありません。

次のMOVEA．L ＃VRAM,A0からアドレスとビット位置の計算を行っています。書き込み開始アドレスを偶数番地にするため，書き込み開始アドレスは，

#VRAM＋Y＊$80＋（X/16＊2）

となります。X方向のドット数を16で割った商を2倍した値を作るという，きりのよい作業を手っ取り早く行うために3ビット右にシフト（8で割るのと同じ）して最下位ビットを0にするという方法を使っています。

図4のNはXを16で割った余りになります。これも割り算するまでもありません。16，すなわち$10で割った余りというのは下位3ビットのデータにほかなりませんから＄FとANDをとって下位3ビットだけを抽出すればよいわけです。これを16から引けば何ビット右にシフトすればよいかが決まります。

1文字単位のマスキング

ところで文字フォントを読み出してそれをシフトして書き込むと，隣に書いてあった文字フォントが消されてしまうことになります。文字を消さないためにはVRAMの内容とORをとればよいのですが，単にそれだけではデータがセットされる一方ですから，文字が次々に重ねられていってしまいます。やはり，文字の幅だけを0でクリアしてからORをとるのが正解でしょう。

＄FFFF0000を先ほど計算した16－Nだけ左にローテートしておくと，文字の位置だけが0で残りが1のマスクパターンができます。シフトではなく，ローテートというところがミソです。

こうして作ったマスクパターンとVRAMとのANDをとれば文字の位置だけが0でクリアされますから，ここでさらにROMのフォントパターンとORをとれば綺麗に文字がはめ込まれるわけです。あとは，VRAMの書き込みアドレスを＄80ずつ足しながら16回行えば16×16ドットのパターンができあがります。

書き込みが終了したら，レジスタの値を回復してRTSで呼び出されたところにリターンして終了です。たいした行数ではありませんから，どうしてもわからない部分はCPUになったつもりでレジスタの値の変化を丹念に追いかけてみればどのように動いているかはすぐにわかるでしょう。

8×8ドットの場合

8×8ドットの場合はputcar8x8からになりますが，16×16のときと大きくは違いません。8×8ドットの場合も，ロングワードの書き込みを行えばまったく同じようにできるのですが，たかだか8ビットを書き込むのに4バイトデータを使うというのもどうかという感じがしたのと，ちょっと違う方法も使ってみたかったので，転送はワード単位で行うことにしました。

このため奇数番地からのワードアクセスができないという68000CPU自身の制約に引っ掛かる場合が出てきます。文字フォントをワード境界をまたぐような場所に書き込むとき，すなわち書き込み始める位置が奇数番地のときはプログラムで2回のバイトアクセスに分割しなくてはなりません。

アドレスが偶数か奇数かは最下位ビットで判断できます。ただし，ビットテスト命令などが使えるのはメモリかデータレジスタに限られており，アドレスレジスタには適用できませんので，一度データレジスタに転送してからビットテストをしています。

最後に

「68000のアセンブラ入門」のつもりが，あまり入門っぽくないと思われるでしょう。定石からいけば，まずは主な命令について説明して，それを組み合わせていくことになるのですが，漢字の書き取りではあるまいし，命令をひたすら覚えるなどという無味乾燥な作業を延々と繰り返すことができる人は少ないでしょう。

一生懸命に命令を覚えたつもりでもいざプログラムを組もうとなると，とたんに行き詰まるのは目に見えています。8080などの初期の8ビットCPUならいざ知らず，68000のような16ビットCPUを実際に使いもしないで一気にすべての命令，すべてのアドレッシングモードとその組み合わせを机上で覚えようなどというのは，正気の沙汰とは思えません。こちらとしても，そんなものを網羅するようなことを考えていては，3，4カ月たってもMOVE命令の説明をしている状態になるのは目に見えています。

そもそもCPUの命令というのは使うためにあるのであって，覚えるためにあるのではありません。使うためにあるものは最低限度のところだけ押さえたら，なにはと

アセンブラプログラムの「読み」方

アセンブラプログラムのとっつきにくさのひとつとして，ニーモニックの読みにくさがあるのではないでしょうか。マシン語というのも外国語のようなものですから，覚えるには音読がいちばんです。ここではリスト5のメインルーチンを例に見てみましょう。

```
34    bsr        init__screen
35    clr.l      -(sp)
36    .dc.w      __SUPER
37    addq.l     #4, sp
38    move.l     d0,user__stack
39
40    move.l     #$41,d0
```

まず，34行。bsrはブランチ・サブルーチン，つまり“init__screen”というラベルのついたサブルーチンに分岐しなさいという命令です。これは相対アドレスでサブルーチンを呼ぶ命令ですが，絶対アドレスで行うものにjsr（ジャンプ・サブルーチン）があります。clr.lはクリア・ロングワード，－(sp)はプリデクリメント・スタックポインタと読んでおきましょう。addやmoveはわかりやすいのですが，後ろにqやm，a，pといった文字がつくことがあります。それぞれ，クイック，マルチプルレジスタ，アドレス，ペリフェラルと読み換えましょう。

.dc.wはメモリ上に定数を確保する疑似命令で，ドット・デファイン・コンスタント・ワードと読むべきなんでしょうが，わかってしまえばアルファベット読みしたほうが誤解が少ないようです。

ときどきオペランド部分で#マークのついたものがありますね。アドレスとして参照するのではなくて，値そのものを参照するという意味ですので，#$41はイミディエイト・ダラー41と読むべきでしょうか（ちなみに$は16進数を表す）。

そのほか，読みにくいと思われるものにbeq，abcd，pea，eoriなどがあります。beqはブランチ・イコール条件分岐命令でイコールの代わりにgt（グレーター・ザン），le（レス・イコールといった条件が指定できます。abcdというシャレみたいなのはアッドBCDで10進加算命令，peaはプッシュ・イフェクティブアドレスで lea（ロード・イフェクティブアドレス）の仲間です。eoriはエクスクルーシブオア・イミディエイトと読みます。Z80ではXORと書いたものが68000ではeorとなっているわけです。

アセンブラのニーモニックは概してハナモゲラになっていますが，それなりに規則性はあるものです。せっかくソースリストが掲載されているのですから，無駄なく読み込んでみていってください。綺麗な命令体系ですからニーモニックの意味がわかれば，マシン語入門はそれほど難しくないはずです。

もあれ使ってみるというのがいちばんで，覚える（人工知能的にいうなら学習する）などということは使っているのが人間である以上，時間の経過とともに自然に行われていくことであると思います。その過程で「こんなことできないかなあ」と思ったら，命令表をめくってみて，目的に近そうな命令があったらとにかく使ってみることです。

慣れてきたら他人の書いたプログラムも読んでみましょう。泥臭いことをやっていたり，思わず膝をポンと叩きたくなるようなうまい方法があったりすることがわかるでしょう。それらの経験やら定石，などが集まって，自分なりのプログラミングスタイルができあがっていくのだと思います。

そのようなわけで，今回はX68000のソフトウェアの階層を，DOSコールからIOCSコール，そしてハードウェアアクセスのレベルと順に降りていきながら，私なりのやり方の紹介をすることにしました。ここ十数年の間にマイクロプロセッサと名のつくものはいろいろと触れてきましたが，初物のコンピュータを扱うときはいつも今回紹介したような制約や手順でやっています。

シングルタスクで動いているパーソナルコンピュータにおけるアセンブラは，システムのあらゆるところをしゃぶり尽くすことができる最終兵器です。Cコンパイラのオブジェクト効率がかかなり高くなり，シングルタスクで使うなら十分なメモリが搭載されるようになった今日，何十Kバイトもある大きなプログラムをアセンブラで作る必要はほとんどないでしょうが，なにかあったときに必殺技として繰り出すことができるようにしておけばX68000の世界がぐっと広がることでしょう。

リスト5

```
*********************************************
*                    リスト５                *
*                                           *
*ハードウェア直接アクセスによる文字表示        *
*                                           *
*                                           *
*     注：コメントの中の>>>や<<<はローテート*
*     を意味します。                          *
*                                           *
*********************************************
_EXIT               equ     $ff00
_SUPER              equ     $ff20
_CRTMOD             equ     $10
_TCOLOR             equ     $15
IOCS                equ     $0f
vram                equ     $00e00000
cgen16x16   equ     $00f00000
cgen8x8             equ     $00f3a000
cgen8x16    equ     $00f3a800
cgen12x12   equ     $00f3b800
cgen12x24   equ     $00f3d000
cgen24x24   equ     $00f40000

            .bss
            .even
user_stack:         .ds.l   1

            .text
            .even

*
* メイン
*
            bsr     init_screen
            clr.l   -(sp)               * スーパーバイザ・モードになる
            .dc.w   _SUPER
            addq.l  #4,sp
            move.l  d0,user_stack

            move.l  #$41,d0         *       character_no = 0x41;

            move.l  #5,d4           *       ycount    = 5;
            move.l  #150,d2         *       yposition = 150;
    line_loop:                      *       do {
            move.l  #150,d1         *               xposition = 150;
            move.l  #3,d3           *               xount = 3;
    culum_loop:                     *               do {
            bsr     putchar24x24    *                       putchar24x24();
            addq.l  #1,d0           *                       character_no++;
            add.l   #24,d1          *                       xposition += 24;

            bsr     putchar12x24    *                       putchar12x24();
            addq.l  #1,d0           *                       character_no++;
            add.l   #12,d1          *                       xposition += 12;

            bsr     putchar16x16    *                       putchar16x16();
            addq.l  #1,d0           *                       character_no++;
            add.l   #16,d1          *                       xposition += 16;

            bsr     putchar8x16     *                       putchar8x16();
            addq.l  #1,d0           *                       character_no++;
            add.l   #8,d1           *                       xposition += 8;

            bsr     putchar12x12    *                       putchar12x12();
            addq.l  #1,d0           *                       character_no++;
            add.l   #12,d1          *                       xposition += 12;

            bsr     putchar8x8      *                       putchar8x8();
            addq.l  #1,d0           *                       character_no++
            add.l   #40,d1          *                       xposition += 40;

            dbra    d3,culum_loop   *               } while (xcount-- > 0);

            add.l   #24,d2          *               yposition += 24;

            dbra    d4,line_loop    *       while (ycount-- > 0);

            move.l  user_stack,-(sp) *      ユーザー・モードに戻る
            .dc.w   _SUPER
            addq.l  #4,sp

            .dc.w   _EXIT
*
* d0 ...... キャラクタ番号
* d1 ...... X座標
* d2 ...... Y座標
*


*
* 24x24ドット・モード
*
putchar24x24:
            movem.l d0-d3/a0-a1,-(sp)

            add.l   #$9b,d0  * $40がきた時にAの文字を出すためのオフセット

            movea.l #vram,a0 * vram = vram_base + y*0x80 + x/8;
            lsl.l   #7,d2    * (mulu #$80,d2)
            adda.l  d2,a0
            move.l  d1,d2
            lsr.l   #3,d2
            adda.l  d2,a0

            move.l  #8,d2               * shift = 8 - (x % 8);
            andi.l  #7,d1
            sub.l   d1,d2

            movea.l #cgen24x24,a1   * cgrom = &cgen24x24[char_number][0][0];
            mulu    #72,d0
            adda.l  d0,a1

            move.l  #$ff000000,d3   * mask = 0xff000000 <<< d2;
            rol.l   d2,d3

            moveq.l #23,d1              * i = 23;

            move.l  a0,d0               * if (vram & 1) == 0) {
            btst    #0,d0
            bne     putchar24x24_odd

    putchar24x24_even:              *       do {
            moveq.l #0,d0           *               fontdata = *cgrom.byte++
;
            move.b  (a1)+,d0125:            lsl.l   #8,d0
*           fontdata <<= 8;
            move.b  (a1)+,d0        *               fontdata += *cgrom.byte+
+;
            lsl.l   #8,d0           *               fontdata <<= 8;
            move.b  (a1)+,d0        *               fontdata += *cgrom.byte+
+;
            lsl.l   d2,d0

            and.l   d3,(a0)         *               *vram.long = (long)*vram
 & mask) | fontdata;
            or.l    d0,(a0)

            adda.l  #$80,a0         *               vram += 0x80;

            dbra    d1,putchar24x24_even  *     } while (i--);

            bra     putchar24x24_end * }
                                    * else {
    putchar24x24_odd:               *       do {
            moveq.l #0,d0           *               fontdata = *cgrom.byte++
;
            move.b  (a1)+,d0
            lsl.l   #8,d0           *               fontdata <<= 8;
            move.b  (a1)+,d0        *               fontdata += *cgrom.byte+
+;
            lsl.l   #8,d0           *               fontdata <<= 8;
            move.b  (a1)+,d0        *               fontdata += *cgrom.byte+
+;
            lsl.l   d2,d0

            and.b   d3,3(a0)        *               *(vram+3) &= mask;
            or.b    d0,3(a0)        *               *(vram+3) |= fontdata;

            ror.l   #8,d3           *               mask >>>= 8;
            lsr.l   #8,d0           *               fontdata >>= 8;

            and.w   d3,1(a0)        *               (int) *(vram+1) &= mask;
            or.w    d0,1(a0)        *               (int) *(vram+1) |= fontd
ata;

            ror.l   #8,d3           *               mask >>>= 16;
            ror.l   #8,d3
            lsr.l   #8,d0           *               fontdata >>= 16;
            lsr.l   #8,d0

            and.b   d3,(a0)         *               *vram &= mask;
            or.b    d0,(a0)         *               *vram |= fontdata;

            ror.l   #8,d3           *               mask >>>= 8;

            adda.l  #$80,a0         *               vram += 0x80;

            dbra    d1,putchar24x24_odd *} while (i-- > 0);

    putchar24x24_end:                       * }
            movem.l (sp)+,d0-d3/a0-a1

            rts
*
* 12x24ドット・モード
*
putchar12x24:
            movem.l d0-d3/a0-a1,-(sp)

            movea.l #vram,a0
            lsl.l   #7,d2    * (mulu #$80,d2)
            adda.l  d2,a0
            move.l  d1,d2
            lsr.l   #3,d2
            andi.b  #$fe,d2
            adda.l  d2,a0

            move.l  #20,d2
            andi.l  #15,d1
            sub.l   d1,d2

```

```
        movea.l #cgen12x24,a1
        mulu    #48,d0
        adda.l  d0,a1

        move.l  #$fffff000,d3
        rol.l   d2,d3

        moveq.l #23,d1

  putchar12x24_loop:
        moveq.l #0,d0
        move.w  (a1)+,d0
        lsr.l   #4,d0
        lsl.l   d2,d0

        and.l   d3,(a0)
        or.l    d0,(a0)

        adda.l  #$80,a0

        dbra    d1,putchar12x12_loop

        movem.l (sp)+,d0-d3/a0-a1

        rts

*
* 16x16ドット・モード
*
putchar16x16:
        movem.l d0-d3/a0-a1,-(sp)

        add.l   #$9b,d0

        movea.l #vram,a0 * vram = vram_base + y*0x80 + (x / 8 * 2);
        lsl.l   #7,d2    *  (mulu #$80,d2)
        adda.l  d2,a0
        move.l  d1,d2
        lsr.l   #3,d2
        andi.b  #$fe,d2
        adda.l  d2,a0

        move.l  #16,d2            * shift = 16 - x % 16;
        andi.l  #15,d1
        sub.l   d1,d2

        movea.l #cgen16x16,a1     * cgrom = &cgen16x16[char_number][0][0];
        lsl.l   #5,d0             * (mulu #$20,d0)
        adda.l  d0,a1

        move.l  #$ffff0000,d3     * mask = 0xffff0000 <<< shift;
        rol.l   d2,d3

        moveq.l #$f,d1            * i = 15;

  putchar16x16_loop:              * do {
        moveq.l #0,d0             *       fontdata  = *cgrom.word++ << shi
ft;
        move.w  (a1)+,d0
        lsl.l   d2,d0

        and.l   d3,(a0)           *           (int)*vram &= mask;
        or.l    d0,(a0)           *           (int)*vram |= fontdata;

        adda.l  #$80,a0           *           vram += 0x80;

        dbra    d1,putchar16x16_loop  * } while (i-- > 0);

        movem.l (sp)+,d0-d3/a0-a1

        rts

*
* 8x16ドット・モード
*
putchar8x16:
        movem.l d0-d4/a0-a1,-(sp)

        movea.l #vram,a0
        lsl.l   #7,d2    * (mulu #$80,d2)
        adda.l  d2,a0
        move.l  d1,d2
        lsr.l   #3,d2
        adda.l  d2,a0

        moveq.l #8,d2
        andi.w  #7,d1
        sub.w   d1,d2

        movea.l #cgen8x16,a1
        lsl.l   #4,d0    * (mulu #16,d0)
        adda.l  d0,a1

        move.w  #$ff00,d3
        rol.w   d2,d3
        move.l  d3,d4

        moveq.l #15,d1

        move.l  a0,d0
        btst    #0,d0
        bne     putchar8x16_odd

  putchar8x16_even:
        moveq.l #0,d0
        move.b  (a1)+,d0
        lsl.l   d2,d0

        and.w   d3,(a0)
        or.w    d0,(a0)

        adda.l  #$80,a0

        dbra    d1,putchar8x16_even

        bra     putchar8x16_end

  putchar8x16_odd:
        moveq.l #0,d0
        move.b  (a1)+,d0
        lsl.w   d2,d0

        and.b   d3,1(a0)
        or.b    d0,1(a0)

        lsr.w   #8,d3
        lsr.w   #8,d0
        and.b   d3,(a0)
        or.b    d0,(a0)

        move.w  d4,d3

        adda.l  #$80,a0

        dbra    d1,putchar8x16_odd

  putchar8x16_end:
        movem.l (sp)+,d0-d4/a0-a1
        rts

*
* 12x12ドット・モード
*
putchar12x12:
        movem.l d0-d3/a0-a1,-(sp)

        movea.l #vram,a0
        lsl.l   #7,d2    * (mulu #$80,d2)
        adda.l  d2,a0
        move.l  d1,d2
        lsr.l   #3,d2
        andi.b  #$fe,d2
        adda.l  d2,a0

        move.l  #20,d2
        andi.l  #15,d1
        sub.l   d1,d2

        movea.l #cgen12x12,a1
        mulu    #24,d0
        adda.l  d0,a1

        move.l  #$fffff000,d3
        rol.l   d2,d3

        moveq.l #$b,d1

  putchar12x12_loop:
        moveq.l #0,d0
        move.w  (a1)+,d0
        lsr.l   #4,d0
        lsl.l   d2,d0

        and.l   d3,(a0)
        or.l    d0,(a0)

        adda.l  #$80,a0

        dbra    d1,putchar12x12_loop

        movem.l (sp)+,d0-d3/a0-a1

        rts


*
* 8x8ドット・モード
*
putchar8x8:
        movem.l d0-d4/a0-a1,-(sp)

        movea.l #vram,a0          * vram = vram_base + y*0x80 + x/8;
        lsl.l   #7,d2             * (mulu #$80,d2)
        adda.l  d2,a0
        move.l  d1,d2
        lsr.l   #3,d2
        adda.l  d2,a0

        moveq.l #8,d2             * shift = 8 - x % 8;
        andi.w  #7,d1
        sub.w   d1,d2

        movea.l #cgen8x8,a1       * cgrom = &cgen8x8[char_number][0];
        lsl.l   #3,d0             * (mulu #8,d0)
        adda.l  d0,a1

        move.w  #$ff00,d3         * mask = mask_buf = 0xff00 <<< shift;
        rol.w   d2,d3
        move.l  d3,d4

        moveq.l #$7,d1            * i = 7;

        move.l  a0,d0             * if ((vram & 1) == 0) {
        btst    #0,d0
        bne     putchar8x8_odd

  putchar8x8_even:                *       do {
        moveq.l #0,d0             *               fontdata  = *cgrom.byte+
+ << shift;
        move.b  (a1)+,d0
        lsl.l   d2,d0

        and.w   d3,(a0)           *               (int)*vram &= mask;
        or.w    d0,(a0)           *               (int)*vram |= fontdata;

        adda.l  #$80,a0           *               vram += 0x80;

        dbra    d1,putchar8x8_even *      } while (i-- > 0);

        bra     putchar8x8_end    * }
                                  * else
  putchar8x8_odd:                 *       do {
        moveq.l #0,d0             *           fontdata = *cgrom.byte++ <<
d2;
        move.b  (a1)+,d0
        lsl     d2,d0

        and.b   d3,1(a0)          *           *(vram+1) &= mask;
        or.b    d0,1(a0)          *           *(vram+1) |= fontdata;

        lsr.w   #8,d3             *           mask >>= 8;
        lsr.w   #8,d0             *           fontdata >>= 8;

        and.b   d3,(a0)           *           *vram &= mask;
        or.b    d0,(a0)           *           *vram |= fontdata;

        move.w  d4,d3             *           mask = mask_buf;

        adda.l  #$80,a0           *           vram += 0x80;

        dbra    d1,putchar8x8_odd *       } while (i-- > 0);
                                  * }
  putchar8x8_end:
        movem.l (sp)+,d0-d4/a0-a1
        rts

*
* 画面の初期化を行います
*
*
init_screen:
        moveq.l #_CRTMOD,d0
        move.w  #16,d1
        trap    #IOCS

        moveq.l #_TCOLOR,d0
        moveq.l #1,d1
        trap    #IOCS

        rts

        .end
```

狙いはスプライト&グラフィック

BASICやC言語で手の届かない部分はどうしてもアセンブラで記述せねばなりません。できるだけ細かな操作がしたいスプライトやグラフィックまわりについてマシン語の基本テクニックを見ていきましょう。

Nakamori Akira 中森 章

アセンブラによるX68000料理教室

X68000マシン語活用へのアプローチとして，前のページではテキスト画面への文字列表示を紹介したわけですが，文字が表示できれば，あとは何をするにも基本は同じです。ここからは，グラフィック，スプライトなどX68000のおいしい部分を中心に攻めていくことにしましょう。ここでもやはり，DOSコールやIOCS コールなどによってシステムサブルーチンのサービスを利用するのが常道です。

幻の画面モードに迫る

最初に紹介するのは，IOCSコールを使ったプログラムです。IOCSコール一覧[1]を眺めているとファンクション番号10Hの「CRTモードの設定」という項が目につきます。X68000では，なんと17種類ものCRTモードを設定することができるのですね。Human68kのDOSコール（0FF23H）では6種類のCRTモードしか設定できませんから，この違いは圧倒的です。

さらに注目すべきはCRTモードの17と18です。通常の使い方ではX68000の最大画面サイズは768×512ドットですが，裏技を使えば1024×424ドット，1024×848ドットで画面表示をすることもできたのです。ただし，現在の専用CRTディスプレイでは24kHzモードでしか表示することができません。これは，将来もっと精度の高いCRTが使えるようになったときのためのモードかもしれませんね。1024×848ドットの画面で絵を描けたら楽しいと思いませんか？ ただし，CZ-603Dや非純正ディスプレイでは動作は保証できませんので注意してください。

そこで，ちょっと実験の意味でこれをプログラミングしてみることにします。絵は単純に800×800ドット程度のボックスフィルにしましょう。通常の画面なら800×800ドットの四角形は画面内に収まりきりませんが，1024×848ドットの画面なら大丈夫です。また，今回のプログラムで必要になるボックスフィルを行うためのIOCSのファンクション番号は0BAH，グラフィック画面の初期化のためのIOCSのファンクション番号は90Hです。

以上，3つのIOCSコール（10H,0BAH, 90H）を用いれば目的のプログラムはあっという間に完成ですね。というわけで，1024×848ドットのサイズの画面にボックスフィルを行うプログラムがリスト1です。リスト1ではIOCSコールを行う共通手順（D0に値を設定してTRAPを実行する）をマクロにしてみました。

プログラムでは，

1) _CRTMOD（IOCSコール10H）
CRTモードをモード18に設定する。
2) _G_CLR_ON（IOCSコール90H）
グラフィック画面をクリアし表示をON にする。
3) _FILL（IOCSコール0BAH）
始点(50, 20)，終点(990, 800)でボックスフィルを行う。
4) _EXIT（ファンクションコールFF00H）
プログラムを終了する。

という手順を順番に行っているだけですから特に説明はいらないでしょう。_CRTMOD の引数はレジスタD1で，_FILLの引数はレジスタA1で渡されていますね。

リスト1のプログラムの実行は写真1に示します（大きな四角）。写真ではわかりにくいと思いますが，1024×848ドットのモードでは画面がかなりちらつきます。しかし，グラフィック画面だけを使うのであればさして問題はないようにも思われます。

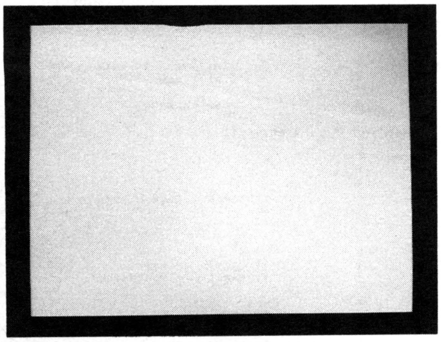


写真1 巨大なボックスフィル

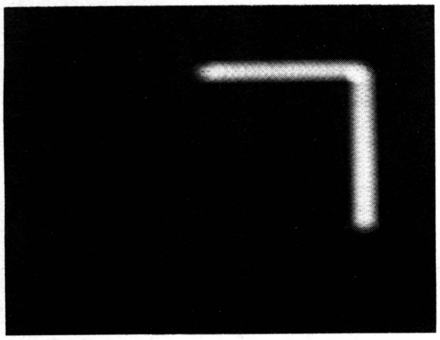


写真2 スプライトを動かす

リスト1 1024×848ドットのグラフィック

```
***************************************
*  ＩＯＣＳコールの使用例（その１）  *
***************************************
_EXIT           equ     $ff00   ; プログラム終了
_CRTMOD         equ     $10     ; ＣＲＴモードの設定
_G_CLR_ON       equ     $90     ; グラフィック画面クリア・ＯＮ
_FILL           equ     $ba     ; ボックスフィル
*
IOCS    macro   number
        moveq.l #number,d0
        trap    #15
        endm
init:
        move.w  #18,d1                  ; 高解像度・1024×848・16色
        IOCS    _CRTMOD                 ; ＣＲＴモードの設定
        IOCS    _G_CLR_ON               ; グラフィック画面クリア・ＯＮ
box_fill:
        lea.l   fill_param,a1
        IOCS    _FILL
fine:
        dc.w    _EXIT

fill_param:
        dc.w    50                      ; 始点Ｘ座標
        dc.w    20                      ; 始点Ｙ座標
        dc.w    990                     ; 終点Ｘ座標
        dc.w    800                     ; 終点Ｙ座標
        dc.w    5                       ; パレットコード
        end
```

IOCSでスプライト

さて，X68000のアセンブリ言語のための究極のメニューともいえるIOCSコールをリスト1だけのプログラムで終わらせるのは惜しいので，もうひとつプログラムを作ってみます。次はスプライトを扱ってみましょう。

アセンブリ言語でスプライトを扱う手順は，X-BASICでの手順と同じで，

1) 画面モードの設定(10H)
2) スプライト画面の初期化(0C0H)
3) スプライトパターンの初期化(0C3H)
4) スプライトパレットの設定(0CFH)
5) スプライトパターンの定義(0C4H)
6) スプライト画面の表示ON(0C1H)
7) スプライトパターンの移動(0C6H)

という手順を踏みます。(　)内はそのためのIOCSコールのファンクション番号です。パターンの移動とは，スプライトレジスタにプレーン番号，パターン番号，X座標，Y座標などを設定することですので注意しましょう。

いきなりプログラムに移りますが，リスト2がIOCSコールを使ってスプライトを扱うプログラムです。プログラムの細かい説明は省略しますが，何をやるプログラムかというと，円形をしたパターンが画面の縁に沿って右回りに移動しながら，画面をひと回りするたびに移動距離を狭め，次第に画面の中央に寄っていくというプログラムです。プログラムの約半分の行数はパターンの移動のための座標計算に使われていて，そこはIOCSコールを使わない純然たるアセンブリ言語のプログラムです。IOCSコールを使わないで書くといかにもアセンブリ言語を使っているという雰囲気になれますね。

なお，リスト2では高解像度，表示画面サイズ512×512ドット，仮想画面サイズ1024×1024ドット，色数16色というCRTモードを使っています。スプライトパターンのデータはX-BASICで定義したパターンをDB.Xで読み出して作りました[2]。プログラムの実行結果を写真2に示しますが，スプライトを扱ったプログラムは実際の動きが見えないと面白くありませんね（元は円のパターンです）。

1) IOCSコールの一覧は，本誌でも1987年7月号で発表されているが，バックナンバーのない方は，『X68000データブック』(小学館)などにも掲載されている。

2) X-BASICのSP_DEF関数でパターンを定義し，X-BASICを抜けてからDB.Xで0EB8000H～0EB807EH番地（パターン0）をダンプしてデータを作った。このアドレスは画面サイズが256×256ドットあるいは512×512ドット以外ではバスエラーが発生してアクセスできないため，DB.Xを起動する前にSCREEN1などのコマンドを実行して画面サイズを変更しておく必要がある。

リスト2　IOCSコールを利用してスプライトを動かす

```
****************************************
*  ＩＯＣＳコールの使用例（その２）  *
****************************************
_EXIT           equ     $ff00   ; プログラム終了
_CRTMOD         equ     $10     ; ＣＲＴモードの設定
_SP_INIT        equ     $c0     ; スプライト面の初期化
_SP_ON          equ     $c1     ; スプライト面の表示ＯＮ
_SP_CGCLR       equ     $c3     ; ＰＣＧのクリア
_SP_DEFCG       equ     $c4     ; ＰＣＧの設定
_SP_REGST       equ     $c6     ; スプライト・レジスタの設定
_SPALET         equ     $cf     ; スプライト・パレットの設定
*
IOCS    macro   number
        moveq.l #number,d0
        trap    #15
        endm
init:
        move.w  #0,d1                   ; 高解像度・512×512・16色
        IOCS    _CRTMOD                 ; ＣＲＴモードの設定
        IOCS    _SP_INIT                ; スプライト面の初期化
        IOCS    _SP_CGCLR               ; ＰＧＧ（パターン）のクリア
pal_def:
        move.l  #14,d7
        move.l  #$0fff,d6
        moveq.l #1,d2                   ; パレットブロック１
pdef_loop:
        addi.l  #$1000,d6
        move.l  #15,d1
        sub.l   d7,d1                   ; パレットコード
        move.l  d6,d3                   ; カラーコード
        IOCS    _SPALET                 ; スプライト・パレットの設定
        dbra    d7,pdef_loop            ; 15回ループ する
sp_def:
        moveq.l #0,d1                   ; パターン番号
        moveq.l #1,d2                   ; 16×16ドット
        lea.l   sp_pat0,a1              ; パターンデータ
        IOCS    _SP_DEFCG               ; パターン０の設定
sp_on:
        IOCS    _SP_ON                  ; スプライト面の表示ＯＮ
sp_move:
        move.l  #16,d6                  ; Ｘ座標
        move.l  #16,d7                  ; Ｙ座標
        movea.l #0,a2                   ; 動いている方向
        movea.l #496,a3                 ; 終わりの座標(dir0)
        movea.l #496,a4                 ; 終わりの座標(dir1)
        movea.l #0,a5                   ; 終わりの座標(dir2)
        movea.l #1,a6                   ; 終わりの座標(dir3)
        moveq.l #3,d5                   ; プライオリティ（３に固定）
disp_loop:
        move.l  #$80000000,d1           ; プレーン番号０（垂直帰線検出）
        move.l  d6,d2                   ; Ｘ
        move.l  d7,d3                   ; Ｙ
        move.l  #$0100,d4               ; パターン０・反転なし
        IOCS    _SP_REGST               ; スプライト・レジスタの設定
*
        move.l  #$80000001,d1           ; プレーン番号１（垂直帰線検出）
        move.l  d6,d2
        addi.l  #16,d2                  ; Ｘ＋１６
        move.l  d7,d3                   ; Ｙ
        move.l  #$4100,d4               ; パターン０・横反転
        IOCS    _SP_REGST               ; スプライト・レジスタの設定
*
        move.l  #$80000002,d1           ; プレーン番号２（垂直帰線検出）
        move.l  d6,d2                   ; Ｘ
        move.l  d7,d3
        addi.l  #16,d3                  ; Ｙ＋１６
        move.l  #$8100,d4               ; パターン０・縦反転
        IOCS    _SP_REGST               ; スプライト・レジスタの設定
*
        move.l  #$80000003,d1           ; プレーン番号３（垂直帰線検出）
        move.l  d6,d2
        move.l  d7,d3
        addi.l  #16,d2                  ; Ｘ＋１６
        addi.l  #16,d3                  ; Ｙ＋１６
        move.l  #$c100,d4               ; パターン０・縦横反転
        IOCS    _SP_REGST               ; スプライト・レジスタの設定
*
        cmpa.l  #256,a3
        ble     fine
        move.l  #$7ff,d0
wait:
        dbra    d0,wait
        move.l  a2,d0
        beq     dir0                    ; 左→右　移動
        cmpi.l  #2,d0
        blt     dir1                    ; 上→下　移動
        beq     dir2                    ; 右→左　移動
dir3:
        subq.l  #1,d7                   ; 下→上　移動
        cmp.l   a6,d7
        bge     disp_loop
        movea.l #0,a2
        adda.l  #16,a6
        bra     disp_loop
dir2:
        subq.l  #1,d6
        cmp.l   a5,d6
        bge     disp_loop
        movea.l #3,a2
        adda.l  #16,a5
        bra     disp_loop
dir1:
        addq.l  #1,d7
        cmp.l   a4,d7
        ble     disp_loop
        movea.l #2,a2
        suba.l  #16,a4
        bra     disp_loop
dir0:
        addq.l  #1,d6
        cmp.l   a3,d6
        ble     disp_loop
        movea.l #1,a2
        suba.l  #16,a3
        bra     disp_loop
fine:
        dc.w    _EXIT

sp_pat0:
        dc.l    $00000000,$00000000,$00000000,$00000001
        dc.l    $00000012,$00000123,$00001234,$00012344
        dc.l    $00012345,$00123456,$00123456,$01234567
        dc.l    $01234567,$01234567,$12345678,$12345678
        dc.l    $00000011,$00011122,$01122233,$12233344
        dc.l    $23344455,$34455566,$45566677,$56677788
        dc.l    $67788899,$788999AA,$789AAABB,$89AABBCC
        dc.l    $89ABCCDD,$89ABCDEE,$9ABCDEEF,$9ABCDEF0
        end
```

もうひとつのDOSコール

MC68000では32ビットの乗除算命令，あるいは浮動小数点演算命令といったものはサポートされていません。X68000ではCコンパイラの登場とともに，数値演算プロセッサにも対応した数値演算用のファンクションコールが拡張されました。

これらのファンクションコールには，通常のDOSコールが0FFxxHというコードを持っているのに対し，0FExxH系列の未実装命令が割り当てられています。アセンブラからの呼び出し方などはまったく同じですから，これらはもうひとつのDOSコールと呼んでいいかもしれません（注意：これらのファンクションコールを使用する場合にはシステムにFLOATn.Xが組み込まれている必要があります)。

もちろん，CコンパイラのためのDOSコールだからといって通常のアセンブリ言語によるプログラムで使っていけないという決まりはありません。また，このDOSコールには，乱数を発生させる機能とか文字列を数値に変換する機能とか，通常のプログラミングで役に立ちそうな機能がたくさんありますから，それを使わないのはもったいない話です。ここでは，この0FExxHのDOSコールを使ったプログラムを作ることを考えます。

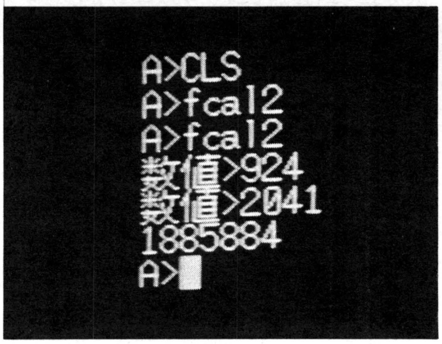


写真3　2つの数値の積

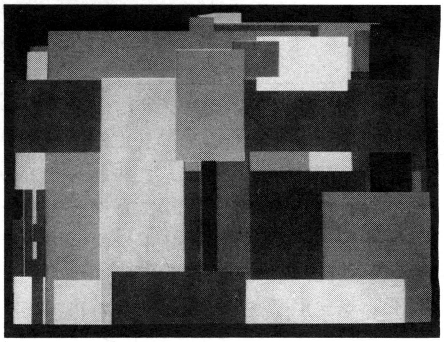


写真4　乱数を使ったボックスフィル

まずは32ビットの乗算を使ってみましょう。32ビットの乗算は0FE00H(__LMUL)という未実装命令で実行できます。これはレジスタD0の内容とレジスタD1の内容の積をレジスタD0に入れるDOSコールです[4]。このDOSコールを使って，2つの数値をキーボードから入力して，その積を表示するというプログラムを作りましょう。問題は，DOSコールの_GETS(0FF0AH)によってキーボードから入力される文字列（数字の並び）を32ビットの数値に変換するプログラムと，DOSコールの_PRINT(0FF09H)で表示するために32ビットの数値を文字列に変換するプログラムです。

この程度のプログラムならアセンブリ言語で書くのに手頃な題材ですが，うまいぐあいに文字列から数値への変換は__STOL(0FE10H)というDOSコール，数値から文字列への変換は__LTOS(0FE11H)というDOSコールでサポートされているので，それを使います。そうしてできたプログラムがリスト3です。文字列を入力して数値に変換するプログラムを2回ループして乗算を行い，その結果を文字列に変換して表示しているだけですから説明はいりませんね。実行結果を写真3に示します。

リスト3のプログラムは文字列の入出力だけで地味ですから，次はもう少し派手なプログラムを作ってみましょう。リスト1に示したプログラムを改造して，もっといろいろなサイズの四角形をもっといろいろな色で塗り潰してみるのはどうでしょうか。つまり，リスト1のプログラムでボックスフィルのための四角形のX座標，Y座標，およびパレットコードを乱数で作ってやるのです。四角形もひとつだけでは面白くありませんから，ループを作ることによってボックスフィルを何回か繰り返しましょう。

そのために必要なのは乱数を発生させるファンクションコールで，それは0FE0EH(__RAND)です。リスト4ができあがったプログラムですが，リスト1とほとんど同

リスト3　32ビットの乗算

```
*****************************************
*  もうひとつのDOSコールの使用例1  *
*****************************************
__LMUL          equ     $fe00   ; d0 ← d0 * d1
__STOL          equ     $fe10   ; d0 ← (a0).10進
__LTOS          equ     $fe11   ; (a0).. ← d0
*
_EXIT           equ     $ff00   ; プログラム終了
_PRINT          equ     $ff09   ; 文字列表示
_GETS           equ     $ff0a   ; 文字列入力
*
start:
        lea.l   NUM1,a6                 ; 数値の退避領域先頭
        moveq.l #1,d7                   ; ループ回数(−1)
num_get:
        pea     PROMPT
        dc.w    _PRINT                  ; プロンプト表示
        addq.l  #4,sp
*
        pea     INPTR
        dc.w    _GETS                   ; 文字列(数値)を入力
        addq.l  #4,sp
*
        pea     CRLF+1
        dc.w    _PRINT                  ; 改行
        addq.l  #4,sp
*
        lea.l   NUMBER,a0
        dc.w    __STOL                  ; 文字列を数値に変換
        move.l  d0,(a6)+                ; 数値を退避
*
        dbra    d7,num_get              ; 数値を2回読む
*
        move.l  NUM1,d0
        move.l  NUM2,d1
        dc.w    __LMUL                  ; 読んだ数値を乗算
*
        lea.l   NUMBER,a0
        dc.w    __LTOS                  ; 数値を文字列に変換
*
        pea     NUMBER
        dc.w    _PRINT                  ; 変換した文字列を表示
        addq.l  #4,sp
*
        pea     CRLF                    ; 復改・改行
        dc.w    _PRINT
        addq.l  #4,sp
*
        dc.w    _EXIT                   ; プログラム終了
        .even
PROMPT: dc.b    "数値>",0
        .even
CRLF:   dc.b    $0d,$0a,0
INPTR:  dc.b    255                     ; 入力最大文字数
        dc.b    0                       ; 実際に入力された文字数
NUMBER: ds.b    255+1                   ; 入力された文字列
        .even
NUM1:
        dc.l    0                       ; 数値1
NUM2:
        dc.l    0                       ; 数値2
        end
```

じですからここでもプログラムの説明は省略します。なお，リスト4の実行結果を写真4に示しておきます。

こんどはI/Oだ!

さて今度は，I/Oポートを直接操作してグラフィック画面を初期化して絵を描くというプログラムを作ってみたいと思います。

グラフィックを扱ううえで主役となるハードウェアは，なんといってもCRTCとビデオコントローラです。これらのハードウェアの内部レジスタに対して値を設定してやることにより画面のサイズを変えたりグラフィック画面の使用ができるようになるのです。そして，それらの内部レジスタに対する値の設定をI/Oポートを通して行ってやればいいのです。

詳しくは別の文献に譲りますが，CRTCの内部レジスタはX68000のアドレス空間の0E80000$_H$～0E8002E$_H$番地および0E80480$_H$番地に，ビデオコントローラの内部レジスタは0E82400$_H$番地，0E82500$_H$番地，0E82600$_H$番地に割り当てられています。また，テキスト画面のパレットは0E82200$_H$番地から，グラフィック画面のパレットは0E82000$_H$番地から割り当てられています。それと，肝心のグラフィックVRAMは0C00000$_H$番地から，テキストVRAMは0E00000$_H$番地から割り当てられています。これらのI/Oポートに対して所定の値をMOVE命令で書き込むわけです。

そして，I/Oポートに設定すべき値がわかればプログラムは完成したも同然です[5]。リスト5がそのプログラムリストになります。このプログラムでは上で述べたI/Oポートのほかに，OSのワークエリアにも値を設定しています。具体的には，グラフィック画面でのウィンドウの設定，テキスト画面でのカーソルの移動範囲の設定，およびカーソル表示のON/OFF，カーソルの画面左上への移動です。これらはグラフィック画面に絵を描くうえで必要なことではありませんが[6]参考までに設定してみました。リスト5のプログラムの実行結果を写真5に示します。

それでは，以下にリスト5のプログラムの手順を説明します。

1)　スーパーバイザモードに移行
I/Oポートはシステム領域にありますからユーザーモードでは参照できません。なんらかの方法でスーパーバイザモードに移る必要がありますがここではIOCSコールを利用しています。

2)　CRTCとビデオコントローラの設定
画面モードを表示画面サイズ512×512ドット，65536色に設定しています。

3)　ウィンドウの設定
画面サイズいっぱい（512×512ドット）に設定しています。

4)　テキスト画面のクリア
同時アクセスモードでプレーンT0（0E00000$_H$～0E1FFFE$_H$）に0を書き込んでいき，同時にプレーンT1,T2,T3もクリアします。

5)　グラフィック画面のクリア
現在は1画面モードなのでスクリーン0（0C00000$_H$～0C7FFFE$_H$）に0を書き込んでいきます。

6)　テキストパレットの初期化
0E82200$_H$番地から始まる16ワードにテキストパレットの初期値をテーブルから引いて設定します。

7)　グラフィックパレットの初期化
0E82000$_H$番地から65536色モードでの変態的なパレットの初期値を設定します。設定方法はIOCSのあるROM内のパレット設定ルーチンを参考にしてあります。

8)　カーソルの移動範囲の設定

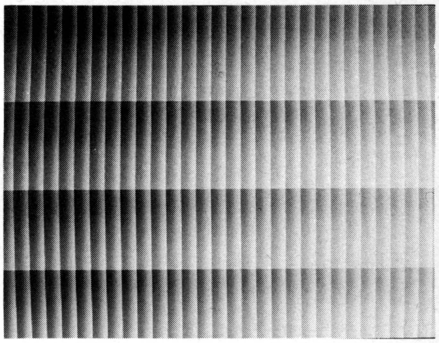


写真5　2×4の四角形を画面いっぱいに

プログラムの実行後，カーソルの移動範囲が512×512ドットの画面いっぱいになるように設定します。また，カーソルの位置を画面の左上に移動します。このと

3) C compiler PRO-68KのシステムディスクのincludeというディレクトリにFEFUNC.Hというファイルがあり，このファイルの中にすべてのファンクションコールの定義がある（EQU疑似命令による名前の定義が並んでいる）。

4)　Cコンパイラでは32ビットの乗算では0FEE0$_H$(__CLMUL)というファンクションコールを用いている。これはスタック上にある2つの32ビット整数の積を計算してスタックに書き戻すファンクションコールである。必ずしもレジスタD0とD1の積を計算するわけでないのでスタックを値の受け渡しに使うようだ。

リスト4　乱数を使ったグラフィック

```
*******************************************
*  もうひとつのDOSコールの使用例2  *
*******************************************
__RAND          equu    $fe0E   ; d0 ← 乱数
*
_EXIT           equ     $ff00   ; プログラム終了
*
_CRTMOD         equ     $10     ; CRTモードの設定
_G_CLR_ON       equ     $90     ; グラフィック画面クリア・ON
_FILL           equ     $ba     ; ボックスフィル
IOCS    macro   number
        moveq.l #number,d0
        trap    #15
        endm
init:
        move.w  #18,d1          ; 高解像度・1024×848・16色
        IOCS    _CRTMOD         ; CRTモードの設定
        IOCS    _G_CLR_ON       ; グラフィック画面クリア・ON
box_fill:
        lea.l   fill_param,a1
        move.l  #128,d7
box_loop:
        dc.w    __RAND
        andi.w  #$3ff,d0
        move.w  d0,(a1)         ; 始点X座標
        dc.w    __RAND
        andi.w  #$3ff,d0
        move.w  d0,2(a1)        ; 始点Y座標
        dc.w    __RAND
        andi.w  #$3ff,d0
        move.w  d0,4(a1)        ; 終点X座標
        dc.w    __RAND
        andi.w  #$3ff,d0
        move.w  d0,6(a1)        ; 終点Y座標
        dc.w    __RAND
        andi.l  #$f,d0
        move.w  d0,8(a1)        ; パレットコード
*
        IOCS    _FILL
*
        dbra    d7,box_loop
fine:
        dc.w    _EXIT

fill_param:
        dc.w    0               ; 始点X座標
        dc.w    0               ; 始点Y座標
        dc.w    0               ; 終点X座標
        dc.w    0               ; 終点Y座標
        dc.w    0               ; パレットコード
        end
```

きカーソル表示がONだと，カーソルを移動したあと，移動前の位置にごみが残るのでカーソル表示をOFFにしてから一連の作業をしています。

9)　絵を描く

それぞれ異なったパレットで塗り潰した2×4ドットの四角形を512×512ドットの画面いっぱいに張りつけます。画面には32768色が表示されるはずです。

10) ユーザーモードに移行

やるべき処理を終えたので終了する前にユーザーモードに戻ります。これもIOCSコールによります。

11) プログラム終了

DOSコール（_EXIT）を使ってプログラムを終了させます。

こうしてみると，スーパーバイザモードへの移行とプログラムの終了を除けば，ほかにDOSコールやIOCSコールを使わなくてもそれなりのプログラムができることがわかるでしょう。ところで，リスト5のプログラムをアセンブルするとワーニング(警告）メッセージがたくさん出てきます。これはOSのワークエリアの指定に直接アドレスを指定したからです。が，どんなにたくさんのワーニングメッセージが出ても，アセンブルが，

No Fatal error(s)

で終了していれば気にすることはありません。

＊　＊　＊

とまあ，グラフィックやスプライトなどを使うといっても，基本的な考え方は文字表示のときと同じであることがおわかりでしょうか。まだまだ，いろいろなサンプル例を紹介したいのですが，ひとまずこのへんで終わりにしたいと思います。さすがにI/Oを直接さわるのは注意が必要ですが，DOSコールやIOCSコールを使う分には安心ですので，あとは皆さんそれぞれがいろいろと挑戦してみてください。

5)　I/Oポートに設定すべき値はそれぞれのICのマニュアルを見なければならない。しかし，CRTCやスプライトICなどのようにカスタム品のICのマニュアルが手に入るとも思えない。一番てっとり早いのはIOCSのROMをDB.Xなどで逆アセンブルしてそれぞれの場合にどのような値をI/Oポートに設定しているかを見ることだ。0FFxx$_H$のファンクションコール用のベクタは1800$_H$番地から，IOCSコール用のベクタは400$_H$番地から格納されている。たとえば，ファンクション番号NのIOCSコールがどういう処理をしているのか知りたければ（400$_H$＋4＊N）番地に格納されているアドレスから逆アセンブルすればよい。なお，0FExx$_H$のファンクションコール用のベクタはシステムの構成によって違いがあるようだ。

6)　グラフィック画面でのウィンドウ設定はIOCSコールなどで絵を描くときには参照されるがグラフィックVRAMに直接絵を描くときには意味を持たない。

リスト5　I/Oを操作して絵を描く

```
*********************************
*  Ｉ／Ｏポートに直接アクセス  *
*********************************
_EXIT               equ     $ff00
_B_SUPER            equ     $81
IOCS      macro    number
          moveq.l #number,d0
          trap     #15
          endm

start:
          suba.l   a1,a1
          IOCS     _B_SUPER          ; スーパバイザーモード
          move.l   sp,ssp_save
crt_mode:
          move.b   #0,$992           ; カーソル表示　ＯＦＦ
          movea.l #$e80000,a0        ; 高解像度・512×512・65536色
          move.w   #$315,$28(a0)     ; CRTC  R20
          move.w   #$05b,(a0)        ; CRTC  R00
          move.w   #$009,$02(a0)     ; CRTC  R01
          move.w   #$011,$04(a0)     ; CRTC  R02
          move.w   #$051,$06(a0)     ; CRTC  R03
          move.w   #$237,$08(a0)     ; CRTC  R04
          move.w   #$005,$0a(a0)     ; CRTC  R05
          move.w   #$028,$0c(a0)     ; CRTC  R06
          move.w   #$228,$0e(a0)     ; CRTC  R07
          move.w   #$01b,$10(a0)     ; CRTC  R08
          move.w   #3,$2400(a0)      ; ビデオ制御レジスタ R1
          move.w   #$2f,$2600(a0)    ; ビデオ制御レジスタ R3
set_window:
          move.w   #0,$968           ; ウインドウ始点Ｘ座標
          move.w   #0,$96a           ; ウインドウ始点Ｙ座標
          move.w   #511,$96c         ; ウインドウ終点Ｘ座標
          move.w   #511,$96e         ; ウインドウ終点Ｙ座標
          jsr      clr_text          ; テキスト画面クリア
          jsr      clr_graph         ; グラフィック画面クリア
          jsr      set_tpalet        ; テキストパレット初期化
          jsr      set_gpalet        ; グラフィックパレット初期化
cursor_range:
          movea.l #$948,a0
          clr.l    (a0)              ; カーソ移動範囲先頭
          move.w   #63,$28(a0)       ; カーソル移動範囲Ｘ方向
          move.w   #31,$2a(a0)       ; カーソル移動範囲Ｙ方向
          clr.l    $2c(a0)           ; カーソルを画面左上に
draw_pict:
          jsr      draw              ; 絵を描く
user_mode:
          move.b   #$ff,$992         ; カーソル表示ＯＮ
          movea.l ssp_save,a1
          IOCS     _B_SUPER          ; ユーザーモード
fine:
          dc.w     _EXIT             ; プログラム終了
clr_text:
          move.w   $2a(a0),d1
          move.w   #$01f0,$2a(a0)    ; 同時アクセスモード
          move.w   #$ffff,d0         ; ($e20000-$e00000)/2-1
          movea.l #$e00000,a1
do_clr_txt:
          move.w   #0,(a1)+          ; テキスト画面クリア
          dbra     d0,do_clr_txt
          move.w   d1,$2a(a0)
          rts
clr_graph:
          movea.l #$c00000,a1        ; 65536色（１画面）モード
          movea.l #$c80000,a2        ; ではグラフィックＶＲＡＭは
do_clr_graph:
          move.w   #0,(a1)+          ; $c00000～$c7fffeのアドレス
          cmpa.l   a2,a1             ; になる
          blt      do_clr_graph
          rts
set_tpalet:
          lea.l    tpal_table,a1     ; テキスト画面の
          movea.l #$e82200,a2        ; パレットの
          move.w   #15,d0            ; 初期値を設定する
do_set_tpalet:
          move.w   (a1)+,(a2)+
          dbra     d0,do_set_tpalet
          rts
set_gpalet:
          movea.l #$e82000,a1        ; 65536色モードでの
          move.w   #$7f,d0           ; パレットの
          move.w   #1,d1             ; 初期値を設定する
do_set_gpalet:
          move.w   d1,(a1)+
          move.w   d1,(a1)+
          addi.w   #$202,d1
          dbra     d0,do_set_gpalet
          rts
draw:
          moveq.l #0,d0              ; グラフィック画面に絵を描く
          moveq.l #0,d5              ; パレット（＝カラーコード）
          moveq.l #0,d6              ; Ｘ座標
          moveq.l #0,d7              ; Ｙ座標
          movea.l #$c00000,a0        ; グラフィック画面先頭
gset_loop:
*
          move.l   d7,d0
          lsl.l    #5,d0
          lsl.l    #5,d0
          add.l    d6,d0
          add.l    d6,d0             ; $400*y + x*2
          move.w   #3,d3
loopy:
          move.w   #1,d4             ; ２×４のＢＯＸを描く
          move.l   d0,d1
loopx:
          move.w   d5,0(a0,d1.l)     ; 点を打つ
          addi.l   #2,d1             ; Ｘ増分
          dbra     d4,loopx
          add.w    #$400,d0          ; Ｙ増分
          dbra     d3,loopy
*
          addi.w   #2,d5             ; カラーコード増分（２）
          addi.w   #4,d7             ; Ｙ座標増分（４）
          cmpi.w   #512,d7
          blt      gset_loop
          moveq.l #0,d7
          addi.w   #2,d6             ; Ｘ座標増分（２）
          cmpi.w   #512,d6
          blt      gset_loop
          rts
          .even
ssp_save:
          dc.l     0                 ; ＳＳＰの退避領域
tpal_table:
          dc.w     $0000,$f83e,$ffc0,$fffe ; テキストパレット
          dc.w     $cda8,$cda8,$cda8,$cda8 ; の初期値
          dc.w     $4022,$4022,$4022,$4022
          dc.w     $4022,$4022,$4022,$4022
          end
```

THE SENTINEL

第77部 高速エディタアセンブラREDA
特別付録 X1版S-OS“SWORD”〈再掲載〉

●S-OS用エディタアセンブラ発表

以前から予告しておいたように，S-OS“SWORD”上で動作する新しいエディタアセンブラが完成しました。

さすがにスクリーンエディタとまではいきませんが，ページスクロールによって，それに近い操作が可能で，カット&ペーストやマークジャンプなどに対応したエディタと高速アセンブラの組み合わせです。

一定量のスペースをタブコード化して格納していますので，オンメモリでもZEDAの2，3倍のテキストがアセンブルできるはずです。また，分割アセンブルもセミオートで実行できるようになりました。

ただし，BASICなどでタブキーの使える機種では行番号の関係でタブ位置がBASICのものとずれていますので，テキストをコンパクトに抑えるためにはタブ位置の補正を行っておいたほうがよいでしょう。X1などでは，ctrl-Yですべてのタブを消去したあとctrl-Tで行番号位置から8文字おきにタブを再設定するようにします。

●X1版“SWORD”特別再掲載

3年半にわたり毎月発表され続けている全機種共通システム用のアプリケーションですが，新しいユーザーのなかにはS-OSを持っていないという人も現れてきましたので，X1版に限り再掲載することになりました。REDAを動かすにも“SWORD”が必要ですから，お持ちでない方はぜひとも入力しておいてください。

今回掲載されているものは基本的に1987年3月号で再掲載されたものと同じです。1987年5月号で瀧山氏によるRAMディスクやバッチ処理などの機能拡張は一切加えられていません。これが“SWORD”のもっとも標準的なかたちですので，あとは必要に応じて拡張してください。

全機種共通システム掲載記事

■85年6月号
序論　共通化の試み
第1部　S-OS“MACE”
第2部　Lisp-85インタプリタ
第3部　チェックサムプログラム
■85年7月号
第4部　マシン語プログラム開発入門
第5部　エディタアセンブラZEDA
第6部　デバッグツールZAID
■85年8月号
第7部　ゲーム開発パッケージBEMS
第8部　ソースジェネレータZING
■85年9月号
インタラプト　S-OS番外地
第9部　マシン語入力ツールMACINTO-S
第10部　Lisp-85入門(1)
■85年10月号
第11部　仮想マシンCAP-X85
連載　Lisp-85入門(2)
■85年11月号
連載　Lisp-85入門(3)
■85年12月号
第12部　Prolog-85発表
■86年1月号
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■86年2月号
第15部　S-OS“SWORD”
第16部　Prolog-85入門(1)
■86年3月号
第17部　magiFORTH発表
連載　Prolog-85入門(2)
■86年4月号
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■86年5月号
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■86年6月号
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ&エディタ
第24部　“SWORD”2000 QD
連載　対話で学ぶmagiFORTH
特別付録　PC-8801版S-OS“SWORD”
■86年7月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版S-OS“SWORD”
■86年8月号
第26部　対局五目並べ
第27部　MZ-2500版S-OS“SWORD”
■86年9月号
第28部　FuzzyBASIC発表
連載　明日に向かってmagiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタDREAM
第31部　FuzzyBASIC料理法〈1〉
■86年11月号
第32部　パズルゲームHOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC料理法〈2〉
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC料理法〈3〉
■87年1月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC料理法〈4〉
■87年2月号
第36部　アドベンチャーゲームMARMALADE
第37部　テキアベ作成ツールCONTEX
■87年3月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツールMAGE
付録　“SWORD”再掲載とMAGICの標準化
■87年4月号
第40部　INVADER GAME
第41部　TANGERINE
■87年5月号
第42部　S-OS“SWORD”変身セット
第43部　MZ-700用“SWORD”をQD対応に
■87年6月号
インタラプト　コンパイラ物語
第44部　FuzzyBASICコンパイラ
第45部　エディタアセンブラZEDA-3
■87年7月号
第46部　STORY MASTER
■87年8月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージJACKWRITE
特別付録　FM-7/77版S-OS“SWORD”
■87年9月号
第49部　リロケータブル逆アセンブラInside-R
特別付録　PC-8001/8801版S-OS“SWORD”
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASICコンパイラの拡張
第52部　X1turbo版S-OS“SWORD”
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OSの仲間たち
第53部　もうひとつのFuzzyBASIC入門
第54部　ファイルアロケータ&ローダ
インタラプト　S-OSこちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo版“SWORD”アフターケア
　　　　ラインプリントルーチン
特別付録　PASOPIA7版S-OS“SWORD”
■88年1月号
第58部　Fuzzy BASICコンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年2月号
第59部　シューティングゲームELFES
■88年3月号
第60部　構造型コンパイラ言語SLANG
■88年4月号
第61部　デバッギングツールTRADE
第62部　シミュレーションウォーゲームWALRUS
■88年5月号
第63部　シューティングゲームELFES II
第64部　地底最大の作戦
■88年6月号
第65部　構造化言語SLANG入門(1)
第66部　Lisp-85用NAMPAシミュレーション
■88年7月号
第67部　マルチウィンドウドライバMW-1
連載　構造化言語SLANG入門(2)
■88年8月号
第68部　マルチウィンドウエディタWINER
■88年9月号
第69部　超小型エディタTED-750
第70部　アフターケアWINERの拡張
■88年10月号
第71部　SLANG用ファイル入出力ライブラリ
第72部　シューティングゲームMANKAI
■88年11月号
第73部　シューティングゲームELFES IV
■88年12月号
第74部　ソースジェネレータSOURCERY
■89年1月号
第75部　パズルゲームLAST ONE
第76部　ブロックゲームFLICK

＊以上のアプリケーションは，基本システムであるS-OS“MACE”またはS-OS“SWORD”がないと動作しませんので注意ください。

全機種共通S-OS"SWORD"要

高速エディタアセンブラREDA

お待たせしました。ZEDAに代わるS-OS用のエディタアセンブラを発表します。今回のマシン語特集で使われていたZ80用アセンブルリストはすべてこのREDAで出力されています。開発，入門用としてぜひ入力してください。

瀧山 孝 *Takiyama Takashi*　進藤 哲哉 *Shindou Tetsuya*

REDA (Rapid EDitor Assembler) はS-OS"SWORD"上で動作するZ80エディタアセンブラです。これまでOh!Xの標準アセンブラとして使われてきたZEDA（EDASM），およびその改良・高速版ZEDA-3と比べて，以下のような長所を持っています。

1)　アセンブル速度がZEDA-3の2～数倍。元祖ZEDAとは比べるのがかわいそうなくらいです。

2)　エディタ部分を切り離し，アセンブラのみでも動作可能です。これにより，テキストやオブジェクトを格納する領域をいくらか大きく確保することができます。

3)　専用エディタとの組み合わせにより，快適なデバッグ環境を提供します。アセンブルエラーが起きると，自動的にエディタに切り換わり，エラーの発生した行/位置を表示しますから，即，修正が行えます。

4)　専用エディタ使用時は，複数のスペースを1バイトのタブコードに置き換えるので，ソースファイルのサイズが小さくてすみます。

5)　分割アセンブルが使いやすいかたちに改良されています。ファイル名を与えるだけで，自動的にロード/アセンブル（必要ならセーブも）します。

6)　いくつかのアセンブルオプションにより，エラーチェックを厳しくすることができます。ORGのつけ忘れによる暴走も回避できます。

7)　ZEDAが掲載された号はすでに入手が不可能でしたが，今月号は「ここ」にあります。

ただ，次のような短所もあります。

1)　プログラムサイズが若干大きくなってしまいました。ですが，アセンブラ単体でも動作可能ですので，すでにE-MATEやWINERをお持ちの方は，とりあえずアセンブラだけを入力して使うことができます。この場合，長所の3)，4)が犠牲になりますから，最終的には専用エディタも入力し，セットで使うことをお勧めします。

2)　ZEDAにあった疑似マクロ命令はサポートしていません。また，マルチステートメントも許していません。

3)　リロケータブルアセンブラでなければマクロアセンブラでもありません。なんの飾りもない2パスのアブソリュートアセンブラです。名前は「レダ」と読んでください。

入力&実行方法

リスト1を各機種のモニタないしはMACINTO-Cなどのマシン語入力ツールから打ち込み，チェックサム，CRCにより打ち込み間違いがないことを確認したうえで3000H～4FFFHの範囲を実行アドレス3000Hでテープ/ディスクにセーブしてください。また，アセンブラのみを使おうという方は3005H番地を00Hに変更したうえで 3000H～45FFHの範囲をセーブしてください。

S-OS"SWORD"のモニタから，

　＃J3000

と入力すると，アセンブラモードでコールドスタートします。また，3003H番地にジャンプすればホットスタートします。コールドスタートとホットスタートの違いは，囲みのとおりです。

エディタ部
〈エディタの使い方〉

REDAのエディタはBASICと同様のカーソルエディタです。最初にアセンブラを起動して，

　：E

でエディタが起動します。

エディタを起動すると行番号が0001と表示され，画面の下にプロンプト">"が表示されてカーソルが点滅を始めます。このような状態をエディタのコマンドモードと呼びます。

アセンブラをホットスタートするか，いったんエディタを抜けたあと再びエディタを起動すると，作成されているテキストが画面に行番号つきで表示され，コマンドモードに入ります。表1にコマンドモードで使用できるコマンドを挙げておきます。

コマンドモード

コマンドモードでは，主にファイルの入出力，文字列の検索/置換，テキストの連続スクロールを行います。主なコマンドの使用法を説明しておきましょう。

●指定した行から表示する

"G"コマンドを入力すると，表示を開始する行番号をたずねてきます。表示を開始したい行番号を入力してください。行番号は1～9999までで，存在しない行番号を指定するとテキストの最後が表示されます。テキストの最後にはなにもありませんので，画面にはプロンプトだけが表示されます。"Z"コマンドで逆スクロールさせると1ページ分バックし，前の画面を見ることができます。

●ファイル入出力

"F"を入力すると，「Load, Save, Get-cut, Cut-save, Dir」と表示されます。Loadは別のテキストのロードです。Saveは現在編集中のテキストのセーブ。Get-cut, Cut-saveは，カットバッファへのテキストの読み込みと，カットバッファのセーブを行います。いずれも先頭の1文字を入力してください。カットバッファについてはエディ

コールドスタートとホットスタートの違い

コールドスタート（J3000）

タイトルを表示したうえで，特殊ワークの大きさに応じてハッシュテーブルの大きさを決める。続いてソーステキスト格納アドレス，オブジェクト生成可能メモリアドレス，ラベルテーブルの大きさ，並びに使用可能なラベルの最大数，さらにオブジェクトをセーブするときのデバイス名を表示する。さらに，専用エディタが組み込まれていれば，テキストエリアの初期化を行う。エディタが組み込まれていなければテキストエリアの初期化は行わない。以上の初期設定がすんだらアセンブラのコマンドモードに入る。

ホットスタート（J3003）

一切のメッセージを出さず，テキストエリアの初期化も行わないで，アセンブラのコマンドモードに入る。

ットモードで説明します。

Get-cutとCut-saveの2つのコマンドは，別のテキストを現在のテキストに読み込むため，現在のテキストの一部をセーブするために用意してあります。使用したファイル名は保存されていますので，ロード/セーブ方法を指示すると，現在覚えているファイル名を表示して変更待ちになります。ファイル名が不明のときはサブコマンドDでディレクトリを表示し，ファイル名を取り込んでからファイル操作をやり直してください。

●文字列の置換/検索

文字列の検索，置換を実行すると，「Begin, Here」と表示されます。テキストの最初から行うのか，現在行から行うのかを選択するわけです。現在行とは，現在，画面のいちばん上の行を指します。頭文字でいずれかを選択すると，最新の検索語が表示され，修正を促します。OKならそのままリターンキーを押してください。置換の場合はさらに最新の置換語が表示されますので，こちらもOKならリターンキーを押してください。検索，置換が始まります。

検索の最中は，リターンキーで次の候補を探しにいきます。置換ではリターンキーは「置換してもいい」という指示を兼ねます。置換したくない場合はリターンキー，ブレイクキー以外のキーを押してください。置換せずに次の候補を探します。どちらのコマンドもブレイクキーで終了します。

置換語，検索語は半角20文字以内で，コントロールコードを含めて検索できます。たとえば「LD《タブ》」を探すなら，

[L] [D] [^] [I]

と入力してください（コントロールコードの表記法はアセンブラの解説を参照)。

エディットモード

コマンドモードの“E”コマンドで入ります。エディットモードに入るとカーソルは画面の最上段の行に移動し，行番号の直後にある空白のところで点滅を始めます。ここはコントロールカラムです。コントロールカラムに表2の文字を書き込むことで，削除や挿入などの編集作業を行います。特に画面スクロール関係は充実していますので，慣れればエディットモードから抜けることなく編集作業のすべてを行うことができるようになるでしょう。

それぞれのコマンドを説明します。

●行の挿入と削除

最初にテキストを書き込むときは，コントロールカラムに“＋”を書いてから書き始めます。リターンキーを押すと，入力した行の「下に」10行新しい行ができます。このエディタでは行番号のないところにテキストを書くことはできませんので，“＋”コマンドを使って新しい行を作成してから，テキストを書き込むというのが基本作業になります。

不必要な行はコントロールカラムに“－”を書いてリターンキーを押すと削除されます。“/”コマンドは，なにも書いてない複数の行を削除するのに使用します。これは主に，“＋”コマンドであけすぎた行を削除するのに使います。とりあえず何行か“＋”コマンドであけておいて，テキストを作った結果，不要な行ができたら“/”で削る，といういい加減なプログラミングが可能なのです。

●行のマークとマークへのジャンプ

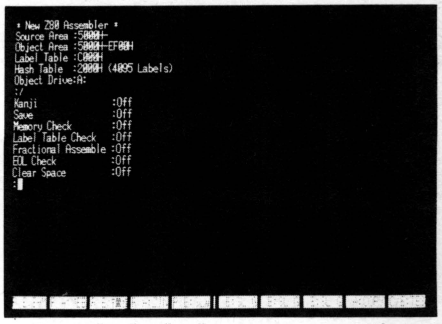
プログラムを作っている最中に，呼び出そうとしているサブルーチンにはどのレジスタで値を渡すのだったかを調べたいなど，ちょっと現在いる行を離れたくなることがよくあります。エディタは行番号のマークという手段でこれをサポートしました。コントロールカラムに“.”をつけてリターンキーを押すと，その行番号をエディタは記憶します。

そして任意の行のコントロールカラムで“,”コマンドを使用すると，マークした行へ帰ってきます。このときエディタは“,”コマンドが実行された行を新しいマーク行として記憶します。ですからここでもう一度“,”コマンドを使用すると，さっきまでながめていた行に飛んでいくことができます。つまり，テキストの2つの場所をいったりきたりできるわけです。

●画面スクロールの制御

プログラムを自由にながめるため，スクロール系のコマンドが充実しています。コントロールカラムに“[”コマンドを書くと，その行から再表示されます。サブルーチンの最初が画面の下部に表示され，全体を見ることができないときに便利でしょう。逆に“]”コマンドは，その行の直前の行を画面最下行として表示し直します。サブルーチンの終わりが画面の上部に表示されたときに便利です。“＝”コマンドは，その行が画面中央になるように再表示します。

“(”コマンドと“)”コマンドは，エディットモードを抜けなくてもページスクロールできるようにしたものです。複数行のスクロールには向きませんが，1，2ページなら楽に使えるでしょう。



残った“<”と“>”の2つのコマンドは，次節で説明しましょう。

エディットモードを終了するには，ブレイクキーまたはシフトブレイクを押します。

●テキストのカットとコピー

エディタで任意の複数行を削除する方法はちょっと変わっています。削除を行う“<”は，「マーク行から“<”が入力された直前の行までを削除する」というコマンドです。削除するには，まず削除を始めたい行をマークします。そして削除最終行の次の行で“<”コマンドを使用するわけです。～行から～行までを削除するという方法では，削除開始行，あるいは削除終了行を覚えておかなければなりません。ここではこの非人間的な方法を排除しました。

削除した行はバッファに保存してあり，次に削除が行われるまで内容が保存されています（もちろん電源を落としたらパー)。削除した行をバッファから取り出すのが“>”コマンドです。エディタには削除のみでコ

表1　コマンドモード

E	エディットモードに入る。カーソルは画面最上行の入力開始位置に移動
V	次の画面を表示
Z	前の画面を表示
G	表示を開始する行番号入力
S	検索（順方向のみ）
C	置換（順方向のみ）
F	ファイルのロード/セーブ
N	編集中のテキストを破棄
R	破棄したテキストを復活
T	テキスト格納アドレスを変更
M	メモリの使用状況をレポート

表2　エディットモード

＋	現在行の後ろに空白行を10行挿入
－	現在行を削除
/	現在行と，続く改行だけの行を削除
.	現在行をマーク
,	現在行とマーク行を交換
[	現在行から表示する
＝	現在行を画面の中心に表示
]	現在行の前の行を最下行に
(	前の画面を表示
)	次の画面を表示
<	マーク行から現在行までを削除
>	削除した行を現在行の前に挿入

ピーが用意されていませんが，コピーはこのカットアンドペーストで対応します。

●テクニカルサポート

コマンドモードとエディットモードはそれぞれが終了されるまでループを続けます。エディタではループを構成するのにちょっと変わった方法を使用しています。それぞれのループの最初でループの先頭アドレスをPUSHしておき，サブルーチンにはJPで飛んでいくのです。

これは決してほめられたプログラミングテクニックではありませんが，プログラムが短くなるという唯一のメリットを持っています。

```
LP0：  CP     'Z'
       JR     NZ, LP1
       CALL   SCRLDWN
       JR     LPEND
LP1：  CP     'V'
       JR     NZ, ～
       CALL   SCRLUP
       JR     LPEND
         ⋮
LPEND：
```

などという面倒なプログラムが，

```
LOOP：LD     HL, LOOP
      PUSH   HL
      CP     'Z'
      JP     Z, SCRLDWN
      CP     'V'
      JP     Z, SCRLUP
        ⋮
```

とあっさり片づきます。

MC68000のように馬鹿デカイメモリを持っているならいいのですが，Z80には64Kバイトしかありません。ループを実現するこの手法は“必要悪”といえると思います。実際のプログラムは「CP～」「JP Z,～」をテーブル参照を使ってさらに短くしています。

なお，エディタ部はアセンブルし直せばアセンブラと独立して動作可能です。ソースプログラムでは，「＃TXTST」というラベルが定義してあり，プログラム先頭のジャンプ命令の次から始まる内部ワークエリアの「＃TXTST」はコメントとして殺してあります。ラベル定義を削除し，ワークエリアの「＃TXTST」を復活させ，好きなアドレスでアセンブルしてください。3000_Hにアセンブルすれば，分離したアセンブラとの併用で，これまでにない大きなプログラムを作成できるようになります。

表3　アセンブラコマンド

A[/]
　メモリ上にあるソーステキストをアセンブルする。「/」が指定された場合は2パス目でアセンブルリストを画面・プリンタに出力する。

A[/]ファイル名[：ファイル名[…]]
　デバイスからソーステキストをロードし，アセンブルする。複数のファイルが指定されたときは，それらを順にアセンブルする。
　SスイッチがON，FスイッチがOFFになっている場合は，1ファイルアセンブルするごとにオブジェクトをセーブするかどうかを確認するメッセージが表示され，ここでYキーまたはリターンキーを押せば，オブジェクトがセーブされる。
　S，Fスイッチがともに ON のときは，すべてのファイルのアセンブルが終了した時点で，セーブするかどうかの確認メッセージが表示される。

S ssss eeee xxxx[oooo]：ファイル名
　ssss$_H$からeeee$_H$までを実行アドレスxxxx$_H$でセーブする。ooooが指定されたときは，oooo$_H$以降のメモリをssss$_H$にロードされるような形式でセーブする（ooooはOFFSETをつけたときに使用する）。

O
　アセンブル後，定義されたラベルの一覧を画面・プリンタに出力する。

?式
　式の値を計算し，表示する。

#
　一度実行するごとに，アセンブルリストおよびラベルの一覧を画面のみに表示するか，画面とプリンタに同時に出力するかどうかが切り換わる。デフォルトはOFF。

/[スイッチ[＋|－][…]]
　アセンブルオプションスイッチの ON/OFF を切り換え，状況を報告する。スイッチには本文で述べた7種類があり，ONにするときは直後に「＋」を，OFF にするときは直後に「－」を置いて指定する。「＋」は省略可能。
　複数の指定を羅列することができ，また，指定したスイッチは再び切り換え直すまで有効。デフォルトは全スイッチOFF。

X[テキスト先頭アドレス]
　ソーステキスト格納先頭アドレスを指定する。デフォルトは5000$_H$番地。

P[オブジェクト領域先頭アドレス[終了アドレス]]
　アセンブル時，オブジェクトを生成してもかまわないメモリ領域の範囲を指定する。Mスイッチが ON の状態でアセンブル中に，このコマンドで指定した以外のメモリ領域にオブジェクトを生成しようとするとエラーが発生する。デフォルトは5000$_H$～＃MEMAX。

D[デバイス名：]
　デバイスのディレクトリをとる。

V[デバイス名：]
　セミオートアセンブル時にオブジェクトをセーブするデバイスの切り換え，および表示を行う。デフォルトはA：。

E
　エディタが組み込まれていれば，エディットモードに入る。

Q
　アセンブラを抜け，呼び出したシステムに復帰する。

アセンブラ部

〈REDAの文法〉

起動時および，エディタからQコマンドで抜けるとアセンブラモードに入ります。アセンブラモードでは表3のコマンドが使えます。すでにメモリ上にアセンブリソースが存在するなら，

　：A

でアセンブルが行われ，マシン語プログラムが生成されます。REDAは2パス構成のアセンブラであり，ソースリストを2度続けて解析します。1パス目はラベルの値を決めるためのもので，2パス目で実際にオブジェクトを生成します。

アセンブル終了後，定義されたラベルの総数と，オブジェクトの開始アドレス・終了アドレス・実行アドレス・生成された先頭アドレスが表示されます。

REDAでのアセンブリソースは，次のような規則に従います。

1)　各行はラベル定義部，命令文，注釈のフィールドから成ります。必要がなければ，このうちのひとつもしくはすべてを省略してもかまいません。

2)　ラベルは行頭に置くことで定義されます。このときラベルであることを明示するために，直後に「：」をつけることが推奨されています。

3)　命令文はZ80のザイログニーモニックか，このアセンブラの疑似命令（表4）です。

4)　命令文は必ず行頭から1文字以上空けて書き始めなければなりません。が，特に「.LIST」と「.NLIST」の2つの疑似命令は行頭に置くことが許されています。

5)　「；」以降は注釈です。また，のちほど述べますアセンブルスイッチにより構文チェックを厳しくしない限り，命令文の直後から行の終わりまでも注釈とみなされ，アセンブラはこれを無視します。

そのほか，細かな点は表5にまとめておきます。

アセンブルオプションスイッチ

REDAにはエラーチェックをより厳しくするものと，モードを切り換えるものとの計7つのアセンブルオプションスイッチがあります。スイッチのON/OFF切り換え，および状態の報告は「/」コマンドで行います。試しに，

　：/

と入力しますと，使用可能なスイッチの一覧とその状態（デフォルトではすべてOFF）が表示されます。スイッチはここで表示された頭文字によって区別します。

スイッチをONにするには頭文字に続け，「＋」を入力します。OFFにするときは「－」です。例を挙げますと，Kスイッチを ON にしたければ，

：/K＋

OFFにしたければ，

：/K－

となります。

●Mスイッチ

オブジェクトの生成先がPコマンドで指定された範囲に収まるかどうかチェックするようになります。誰にでもORGをつけ忘れてオブジェクトが0000H番地から生成されてしまい，システムを破壊して暴走したという経験があるでしょう。このスイッチをONにしておけば，変なアドレスにオブジェクトを生成しようとした時点でエラーが出て止まりますから，安全に作業することができます。

●Lスイッチ

ラベルテーブルがオーバーフローしたかどうかチェックするようになります。REDAは特殊ワークをラベルテーブルに使っていますから，定義できるラベルの最大数はその大きさに依存します。標準の状態ではラベルテーブルのあふれをチェックしていませんが，Lスイッチは，そのチェックを行うようにするものです。

なお，REDAではラベル検索を高速化するためにハッシュテーブルを導入しており定義できるラベルの最大数はこのハッシュテーブルの大きさにも制限されます（起動時に表示される値がそうです）。Lスイッチの状態にかかわらず，ハッシュテーブルのあふれは常にチェックされています。

●Eスイッチ

テキスト行末にコメント以外の文字列があるかどうかチェックするようにします。REDAではマルチステートメントが許されていませんので，ZEDAのテキストをその

表6　エスケープシーケンス

```
^@   =   00H
^A   =   01H
^B   =   02H
     :
^Z   =   1AH
^[   =   1BH
^"   =   22H
^'   =   27H
^^   =   5EH
```

注）＾Aは[^][A]のように2文字で表記する

表4　疑似命令

ORG　nn

アセンブルするマシン語プログラムの先頭アドレスを指定する。基本的に各ソーステキストの先頭で必ず宣言する必要があり，指定がないときは0000Hからオブジェクトを生成する。

例）　ORG　8000H

なお，ORGはひとつのファイル中に複数置くことができるが，アドレスをさかのぼる宣言は認められていない。

```
例）　ORG　8000H
        :
      ORG　7000H
```

はエラー

OFFSET　nn

アセンブルしたオブジェクトをORGで指定したアドレスとは別の領域に生成するときに使用し，nnで生成アドレスと先頭アドレスとの差を指定する。次の例では「3000H番地にロードされたとき動作するプログラム」を8000H番地以降に生成する。

```
例）　OFFSET　8000H－3000H
      ORG　　3000H
```

なお，OFFSETは各ソースにただひとつのみ存在し，すべてのORGに先立って宣言されなければならない。

EQU　nn

ラベルに値を定義する。nnにラベルを使う場合は，そのラベルの値がすでに確定していなければならない（ただし，2パス構成のアセンブラであるため，ラベルの前方参照は1段階に限り許される）。

例）　#PRINT　EQU　1FF4H

DEFB　n　または　DB　n

1バイト単位の定数をそのままオブジェクトとして生成する。「,」または「：」で区切って複数並べることができる。

例）　DEFB　0, 34H, 'A'：“B”

DEFB　'文字列'　または　DB　'文字列'

「'」または「"」で囲まれた文字列を，先頭から順に1バイトずつオブジェクトとして生成する。文字列の中では表6のエスケープシーケンスが使用できる。「,」または「：」で区切って複数並べることができ，また，DEFB n の形式との混在も可能。

```
例）　DEFB　'TEST'
      DEFB　'^'TEST^'^M^@'
      DEFB　27H, 'TEST', 27H, 13H,00H
```

DEFW　nn　または　DW　nn

2バイトの定数を下位バイト，上位バイトの順に，そのままオブジェクトとして生成する。「,」または「：」で区切って複数並べることができる。

例）　DEFW　1234H, 5678H

DEFS　nn　または　DS　nn

nnバイトのメモリ領域を確保する。nnは未定義のラベルであってはならない。

標準の状態では確保したメモリの値は不定だが，アセンブルオプションスイッチCをONにすることで，0で埋めるように強制することができる。

例）　WORK：　DEFS　2

DEFM　'文字列'　または　DM　'文字列'

「'」または「"」で囲まれた文字列を，先頭から順に1バイトずつオブジェクトとして生成する。文字列の中では表6のエスケープシーケンスが使用可能。

例）　DEFM　'TEST'

.NLIST

A/でアセンブル中のみ有効。アセンブルリスト出力をOFFにする。この命令の置かれた行から次に「.LIST」に出合うまでの範囲はアセンブルリストが出力されない。

なお，この命令と，次項の「.LIST」は，特に先行するスペースなしに，行頭から記述することが許される。

```
例）　.NLIST
   .NLIST
```

.LIST

.NLISTで禁止したアセンブルリスト出力を再開する。

```
例）　.LIST
   .LIST
```

.LIST　nn

アセンブルリスト出力を再開すると同時に，出力する行番号をnnにつけ換える。この命令の次の行の行番号がnnとなる。

```
例）　.LIST　100
   .LIST　100
```

表5　REDAの文法

◎数値

定数またはラベルに定義された値，およびそれらからなる式。

◎ラベル

先頭が数字，「$」，「.」ではなく，スペース，タブ，「：」，「,」，「＋」，「－」，「＊」,「/」,「%」,「'」;「"」を含まない任意の半角文字からなる255文字までの文字列をラベルとして認め，全文字識別する。

◎定数

10進定数，16進定数，2進定数，文字定数，および，特殊な定数として$（その時点のロケーションカウンタ）。

◎10進定数

0～9の数字。

◎16進定数

「$」から始まる16進数，または，0～9で始まり「H」で終わる16進数。

```
例）　$ABCD
      0ABCDH
```

◎2進定数

「%」から始まる2進数，または，「B」で終わる2進数。

```
例）　%01011001
      01011001B
```

◎文字定数

「'」または「"」で囲まれた1文字のASCIIコード。表6のエスケープシーケンスが使用可能。

◎そのほかの定数

単独の「$」はそれが置かれた行のアドレスを表す定数として用いることができる。

```
例）          DJNZ　$
                ‖
      LABEL：　DJNZ　LABEL
```

◎式

加減乗除，および剰余(%)が使用できる。

（小）　＋，－　＜　＊，/，%　（大）

の優先順位に従って計算される。カッコは使用できない。また，オーバーフローは無視される。

まま使うときは，このスイッチをONにして，マルチステートメントがないことを確認するのがよいでしょう。

●Cスイッチ

DEFSで確保した領域を0で埋めるようになります。DEFSで確保した領域はプログラムで初期化するのが自然だと考え，REDAでは0でクリアしないようにしてあります。もし，領域が0でクリアされているのを前提で作られたプログラムがありましたら，CスイッチをONにしてアセンブルしてください。

●Kスイッチ

文字列中に全角文字を使用するとき指定します。エスケープ文字として「^」を採用した結果，シフトJISコードの2バイト目と「^」を混同する可能性が出てきてしまいました。S-OSでは漢字はオプション扱いではありますが，REDAはなにもS-OS用のプログラムを作るためのみにあるわけではありませんから，漢字が使えないというのも困ります。というわけで，このKスイッチが追加されました。メッセージに全角文字を使う場合には，忘れずにこのスイッチをONにしてください。

●Sスイッチ

アセンブル後，生成されたオブジェクトをセーブするかどうか聞いてくるようになります。ただし，メモリ上のテキストをアセンブルした場合は無効です。

●Fスイッチ

断片アセンブルモードにします。これについては次項以降で説明します。

分割アセンブル

REDAはZEDA-3に見られたようなかたちの分割アセンブル，つまり複数ファイルの連続アセンブルをサポートします。これにより，メモリ量の制約なしに，大きなプログラムを作成することができます。ZEDA-3では手作業でファイルのロード/アセンブルを実行していましたが，REDAではそのあたりのことはプログラム側で処理しています。半面，ファイルの分割はコマンドラインの1行で指定できる数までに制限されています。

分割アセンブルは次のような手順で行います。

1)　ソースリストを適当な大きさに分割し，セーブしておきます。ここではAAA,BBBの2つのファイルに分割したとします。OFFSETを指定し，ソースを破壊しつつアセンブルすることを前提にすれば，それぞれのファイルはメモリ量ギリギリの大きさにとることができます。

2)　分割した1番目のリストの最後に適当なラベルをつけます。ここでは仮にAENDとします。

3)　2番目のリストの先頭にORGと必要であればOFFSETをつけ足します。このときORGアドレスとして，1番目のリストの最後につけたラベルを使用します。例の場合ですと

　　ORG　AEND

となります。

4)　同様の作業をすべてのリストに対して行い，セーブし直します。なお，テープにセーブする場合は，すべてのソースは1本のテープに「アセンブルする順番」にセーブしておく必要があります。

5)　ここで

　　：/F－

によって，アセンブルスイッチFをOFFにしておきます。また，ソースを破壊しつつアセンブルする場合には，ここでアセンブルオプションスイッチSをONにし，ファイルひとつごとにオブジェクトをセーブするモードにしておく必要があります。つまり，連続してロードされるソースファイルによって，生成されたオブジェクトが上書きされてしまう前にセーブできるようにするのです。指定の方法は，

　　：/S＋

です。

6)　REDAのアセンブルモードで，

　　：A

に続けてファイル名を1番目から順にコロンで区切って並べます。いまの場合ですと

　　：A AAA：BBB

となります。

7)　次々にファイルが読み込まれ，アセンブル（1パス目）されます。ソースファイルを2度読みする必要上，テープ使用時にはパス1が終了した時点でメッセージが出ますので，テープを巻き戻し，なにかキーを押してください。

8)　再び，ファイルが順次読み込まれ，アセンブル（2パス目）されます。SスイッチがONのときは，ファイルひとつのアセンブルがすむごとにセーブするかどうかの確認メッセージが出ますので，セーブしたければ“Y”を入力してください。すると，Vコマンドで指定されたデバイスに，ソースファイル名の拡張子を「.OBJ」にした名前でオブジェクトがセーブされます。

なお，セーブするデバイスがテープ・QDの場合は，メッセージに従って（必要であれば）テープ・QDの交換を行ってください。

断片アセンブル

また，REDAでは以上のZEDA-3方式に加え，断片アセンブルなるモードをサポートします。感じとしては，このモードのほうが「分割」アセンブルという言葉のイメージに近いのですが，すでにある用語と混乱しても困りますので，断片アセンブルと

表7　アセンブルエラー

◎Illegal Opecode Error
　オペコードが正しくない（ミスタイプか，さもなければ注釈の頭に「;」をつけ忘れたか）
◎Illegal Operand Error
　オペランドが正しくない（ザイログニーモニックにはEX HL，DEはないし，SUB HL，BCもない）
◎Undefined Label Error
　未定義のラベルを使った
◎Redefinition Error
　ラベルの2重定義を行った
◎Illegal Label Error
　ラベルに使用できない文字を使った（特に「%」と「.」に注意）
◎Too Many Labels Error
　ラベルテーブルもしくはハッシュテーブルがいっぱいになった
◎Missing Label Error
　ラベルが置かれるべき位置にラベルがない（行頭にいきなりコロンがあるとか）
◎Too Far Error
　相対分岐可能な範囲外に分岐しようとした
◎Illegal Expression Error
　式が正しくない（数値としては使えない文字が含まれている可能性が高い。また，除算時の分母が0または未定義のラベルの場合にも発生する）
◎Missing [)] Error
　カッコが閉じていない
◎Missing [,] Error
　あるべきはずのカンマがない
◎Missing Quote Error
　文字列が閉じていない（クォーテーションマークのつけ忘れか，そうでなければエスケープシーケンスに誤りがあるかもしれない）
◎Illegal ORG Error
　不正なORGがある（アドレスをさかのぼるORG宣言は認められていない）
◎Illegal OFFSET Error
　不正なOFFSETがある（OFFSETはソース中でただひとつ，すべてのORGに先立って宣言されなければならない。ただし，分割アセンブル時にはひとつのファイルでそれぞれのOFFSETが指定可）
◎Memory Error
　決められた範囲外のメモリにオブジェクトを生成しようとした（MスイッチがONのときのみ発生。ORGやOFFSETのつけ忘れが考えられる）
◎Syntax Error
　そのほかの文法上エラー

いう言葉をでっちあげました。これは次の手順で行います。

1)　ソースリストを適当な大きさに分割し，セーブしておきます。このとき，ひとつのファイルの大きさは「メモリの空き領域－生成されるオブジェクトの大きさ」までに制限されます。つまり，最終的に生成されるマシン語プログラムを格納しておけるだけのメモリを確保しておかなければなりません。また，ソースを破壊しながらのアセンブルは行えません。

2)　アセンブルオプションスイッチFをONにします。具体的には，

　　：/F＋

とします。

3)　：Aに続けてファイル名を１番目から順にコロンで区切って並べます。

4)　次々にファイルが読み込まれ，アセンブルされます。

5)　SスイッチがONのときは，全ソースのアセンブルが終了した時点で，オブジェクトをセーブするかどうか確認してきますので，セーブするのであればYまたはリターンキーを，しないのであればNキーを押してください。セーブする場合のファイル名は最初のソースのファイル名の拡張子を「.OBJ」にしたものになります。

REDAのカスタマイズ

REDAは起動時にアセンブルオプションスイッチなどの初期化を行いません。これを利用して，一度REDAを立ち上げ，スイッチの設定を行ってからセーブし直すことで，容易にデフォルトの環境を変更することができます。設定可能なのは次の５項目です。

1)　アセンブルオプションスイッチ(/コマンド)

2)　テキスト格納アドレス（Xコマンド）

3)　オブジェクト生成可能アドレスの下限値（Pコマンド）

4)　オブジェクトをセーブするデバイス名（Vコマンド）

5)　プリンタのON/OFF(＃コマンド)

また，メモリ番地を直接書き換えることで，以下の設定も可能です。

1)　エディタを組み込むかどうか(3005H番地：00Hなら非組み込み，FFH なら組み込み)。

2)　オブジェクトセーブ時の拡張子(3023H番地からの３バイト：拡張子をASCIIコードで並べる。３バイトに満たないときは残りをスペースで埋める)。

開発を終えて

REDAは純粋に道具として使う目的で作成されました。それと同時に，マシン語入門者にとっても扱いやすいよう配慮したつもりです。ZEDAのような疑似マクロがあったほうが入門者にとってはありがたかったのではないかという意見もあるでしょうが，私はそれに関しては逆の意見です。入門者こそ，生のニーモニックで細部でのレジスタの値やフラグの変化を感じ取りながらプログラムを作っていくべきだと考えます。

それはともかく，最近は手に入れることもできず，高速化だのなんのといいながらパッチ当てだらけにされてしまった（みんな私が悪いんです）ZEDAに代わって，新しいアセンブラをいちから制作し，発表することができたのは嬉しいかぎりです。

アセンブラはツールでしかありません。あとは読者の皆さんの力で使いこなしてやってください。

付録 ZEDAとの互換性

ZEDAのテキストを利用するときは以下の点に注意してください。

1)　疑似マクロ

ZEDAでは使えたIFやそのほかの疑似マクロをREDAはサポートしていません。正しいザイログニーモニックに展開してください。

```
例)    IF    A=0 JR TEST
                ↓
       OR    A
       JR    Z, TEST

       SUB   HL, DE
                ↓
       OR    A
       SBC   HL, DE
```

2)　マルチステートメント

REDAではマルチステートメントを認めていません。マルチステートメントはすべて展開してください。なお，アセンブルオプションスイッチEをONにすることで，マルチステートメントが行われているかどうかをチェックすることができます。

3)　ラベルに使える文字

「%」が使用できなくなりました。また，ラベルの先頭に「.」は使えません。

4)　エスケープシーケンス

文字列中の「^」はエスケープ文字として機能します。文字列のなかで「^」を使っている場合は，それを「^^」に変更してください。

```
例)    DEFM    '^v'
                ↓
       DEFM    '^^v^^'
```

5)　演算子の優先順位

ZEDAでは演算子に優先順位がありませんでしたが，REDAは一般的な演算子の優先順位に従って式の評価を行います。そのため，場合によっては式を変形しないとZEDAと同じ値が得られないことがあります。

例)　1＋2＊3は
　　ZEDAでは(1＋2)＊3＝9
　　REDAでは1＋(2＊3)＝7
　　となります。

リスト１　REDAダンプリスト

```
3000 CD 37 30 18 22 FF 00 00 : 6D
3008 00 50 C3 72 44 00 50 FF : 18
3010 FF 00 00 41 3A 23 23 23 : E3
3018 23 23 23 23 23 23 23 23 : 18
3020 23 23 2E 4F 42 4A 00 ED : 3C
3028 73 C6 45 ED 7B C6 45 CD : BE
3030 D6 1F CD C5 30 18 F4 CD : 90
3038 65 44 0C 2A 20 4E 65 77 : 29
3040 20 5A 38 30 20 86 72 20 : 1A
3048 2A 0D 00 3A 05 30 B7 C4 : 21
3050 17 46 2A 6A 1F 22 0F 30 : 71
3058 CD 2D 32 CD EE 1F CD FE : D1
3060 31 CD 65 44 0D 80 85 20 : D9
3068 3A 00 2A 68 1F CD BE 1F : 95
3070 7C 11 E0 FF 19 22 E0 45 : CC
3078 21 00 04 FE 20 38 0E 26 : AF
---------------------------------
SUM: F6 AE 69 63 67 59 6A FF 1FA0

3080 08 FE 40 38 08 26 10 FE : BA
3088 80 38 02 26 20 22 D7 45 : 3E
3090 CD 65 44 48 0D 48 61 73 : E7
3098 68 85 20 20 3A 00 CD BE : F2
30A0 1F 7C 3D 32 DF 45 CB 3C : 35
30A8 CB 1D 2B 22 D9 45 CD E2 : 02
30B0 1F 48 20 28 00 3E FF CD : B9
30B8 5D 42 CD 65 44 8C 73 29 : 3D
30C0 0D 00 C3 43 31 CD EB 1F : 1B
30C8 3E 3A CD F4 1F ED 5B 76 : 16
30D0 1F CD D3 1F 1A FE 3A C0 : F0
```

```
30D8 CD A5 3F B7 C8 FE 41 38 : A7
30E0 02 E6 DF 08 CD A5 3F 08 : 88
30E8 21 FB 30 35 34 C8 BE 23 : 5E
30F0 28 04 23 23 18 F5 4E 23 : F0
30F8 46 C5 C9 51 23 31 41 C0 : 7A
---------------------------------
SUM: EB 99 98 65 D9 2D 6C 23 642F

3100 32 4F 6C 32 2F 59 31 3F : 17
3108 40 32 4A 29 31 50 EC 31 : 83
3110 58 25 32 53 F8 33 44 A9 : 1A
3118 34 56 36 31 23 D5 31 45 : 5F
3120 2E 31 00 ED 7B C6 45 B7 : 89
3128 C9 CD B2 1F D8 E9 3A 05 : 67
3130 30 B7 C8 C3 1A 46 1A B7 : A3
3138 28 09 CD A3 1F 3A 5D 1F : 76
3140 32 13 30 CD 65 44 88 44 : B7
3148 72 69 76 65 3A 00 3A 13 : 3D
3150 30 CD F4 1F 3E 3A C3 F4 : 3F
3158 1F CD A6 3F E6 DF 01 08 : 9F
3160 00 21 AB 31 ED B1 20 22 : DD
3168 0C 41 AF 37 17 10 FD 4F : A6
3170 13 21 12 30 1A FE 2D 28 : E3
3178 0B FE 2B 28 01 06 13 79 : EF
---------------------------------
SUM: 6A 51 3C A1 E9 02 6B 55 3101

3180 B6 77 18 D5 13 79 2F A6 : 7B
3188 18 F7 3A 12 30 4F 11 E3 : CE
```

```
3190 44 06 08 1A B7 28 0A CD : 22
3198 72 44 CD BA 31 CD EE 1F : 48
31A0 21 CB 19 EB CD 56 42 EB : 40
31A8 10 E9 C9 43 20 45 46 4C : FC
31B0 4D 53 4B 3A EC 45 E6 7F : BB
31B8 3D C9 C5 06 14 CD DF 1F : B0
31C0 C1 CD E2 1F 3A 4F 00 3E : 56
31C8 6E CB 19 DA F4 1F CD E2 : EE
31D0 1F 66 66 00 C9 CD E2 1F : 82
31D8 50 72 69 6E 74 65 72 20 : 04
31E0 4F 00 3A 11 30 2F 32 11 : 3C
31E8 30 4F 18 DB CD B2 1F 38 : 48
31F0 0D 22 0D 30 CD A6 3F FE : 1C
31F8 2D 20 03 CD A5 3F CD B2 : 80
---------------------------------
SUM: 96 89 45 79 F2 D0 03 A2 CB4E

3200 1F 38 03 22 0F 30 CD 65 : ED
3208 44 88 41 72 65 61 20 3A : 9F
3210 00 2A 0D 30 CD BE 1F CD : DE
3218 39 32 2A 0F 30 CD BE 1F : 7E
3220 3E 48 C3 F4 1F CD B2 1F : FA
3228 38 03 22 08 30 CD 65 44 : 0B
3230 87 89 00 2A 08 30 CD BE : FD
3238 1F CD E2 1F 48 2D 00 C9 : 2B
3240 AF 32 F4 45 3D 32 EB 45 : B9
3248 3C D5 EB BE 23 20 FC 2B : 24
3250 36 0D D1 CD E6 3F 69 60 : CF
3258 E5 CD 5D 42 E1 3E 28 CD : 65
3260 F4 1F CD BE 1F CD E2 1F : 8B
```

```
3268 48 29 00 C9 CD 43 44 2A : B8
3270 D7 45 CD 94 1F B7 28 45 : C0
3278 CD F4 1F 23 CD 94 1F FE : 81
---------------------------------
SUM: 9E 1F 08 68 0F 3D 93 9E 84D5

3280 0D 20 F5 23 3E 3A CD F4 : 7E
3288 1F CD 94 1F 23 5F CD 94 : 82
3290 1F 23 57 EB CD BE 1F CD : FB
3298 18 20 7D C6 13 6F 26 00 : 23
32A0 01 14 00 CD A0 42 CD 86 : 17
32A8 42 45 EB CD B5 32 CD C7 : BA
32B0 1F BD 32 18 BD 3A 5C 1F : 98
32B8 3D B8 D2 DF 1F C3 EE 1F : 95
32C0 AF 32 F1 45 32 EC 45 3A : B4
32C8 05 30 32 F4 45 1A D6 2F : BF
32D0 20 05 3D 32 F1 45 13 CD : AA
32D8 A6 3F B7 CA B2 34 ED 53 : 8C
32E0 ED 45 0E 00 0C CD 5D 34 : AA
32E8 F5 3A 5D 1F CD 5F 44 20 : 3B
32F0 02 CB F9 F1 20 EE 79 32 : 70
32F8 EC 45 CD 0B 35 CD 20 33 : 5E
---------------------------------
SUM: 4C 33 94 D4 BA 9D 18 22 89BC

3300 CD 2D 35 20 0A 3A EC 45 : C4
3308 E6 80 C4 EE 33 18 EE 3A : 8B
3310 12 30 F6 ED 3C C0 CD 5D : 4B
3318 33 CD EE 1F D8 C3 87 33 : 62
3320 CD 3B 35 ED 5B ED 45 18 : CF
3328 0C 3A 12 30 E6 10 CC 4F : 99
3330 35 ED 5B EF 45 1A B7 C8 : 4A
3338 CD 7A 34 CD 61 35 3A EB : 03
3340 45 B7 28 E5 CD C0 34 3A : 04
3348 12 30 CB 4F 28 DB E6 10 : 55
3350 20 D7 CD 5D 33 CD EE 1F : 2E
3358 D4 8E 33 18 CC CD 65 44 : EF
3360 8A 3F 00 CD 74 33 FE 0D : 48
3368 C8 E6 DF FE 59 C8 FE 4E : F8
3370 20 F1 37 C9 CD 21 20 FE : 1D
3378 1B C0 CD E2 1F 0D 42 72 : 6A
---------------------------------
SUM: AB A8 89 12 E5 7F FB A1 88E5

3380 65 61 6B 00 C3 2B 30 ED : 3C
3388 5B ED 45 CD A3 1F DD 22 : 1B
3390 E4 45 CD B3 31 28 0D 3A : 49
3398 12 30 E6 10 20 06 21 FA : 79
33A0 1F 22 6E 1F 2A 74 1F 11 : 9C
33A8 15 30 06 0D 23 7E FE 0D : 04
33B0 20 02 3E 20 12 13 10 F4 : A9
33B8 11 13 30 1A CD 5C 44 20 : FB
33C0 6A B7 CD C9 33 CD 2B 34 : 16
33C8 37 D9 F5 CD E2 1F 53 65 : 8B
33D0 74 20 00 F1 11 D3 44 30 : DD
33D8 03 11 CB 44 CD E5 1F CD : C1
33E0 E2 1F 54 61 70 65 00 D9 : 64
33E8 CD 74 33 C3 EE 1F 11 DB : 30
33F0 44 18 E9 CD 33 20 18 8C : 09
33F8 CD 57 34 D8 22 70 1F 22 : 03
---------------------------------
SUM: F3 ED 76 8A 89 91 D5 6D A10D

3400 E2 45 4D 44 CD 57 34 D8 : E8
3408 B7 ED 42 23 22 72 1F CD : 89
3410 57 34 D8 22 6E 1F CD A6 : 85
3418 3F FE 3A 28 0D CD B2 1F : 4A
3420 D8 22 E2 45 CD A6 3F FE : D1
3428 3A C0 13 3E 01 CD A3 1F : DB
3430 CD AF 1F DA F3 33 CD 65 : CD
3438 44 57 72 69 74 8B 00 CD : 42
3440 38 44 2A E2 45 22 70 1F : 7E
3448 CD AC 1F 38 E6 CD E2 1F : 84
3450 44 6F 6E 65 0D 00 C9 CD : 29
3458 A6 3F C3 B2 1F 3E 04 C5 : 80
3460 CD A3 1F C1 1A FE 3A 20 : C2
3468 01 13 CD A6 3F B7 C9 CD : 13
3470 09 20 38 BF C8 CD 38 44 : 31
3478 18 F5 CD 5D 34 ED 53 EF : 9A
---------------------------------
SUM: 30 B5 92 2B 4B 82 2E A9 0D84

3480 45 3A EB 45 B7 28 04 CD : 5F
3488 B3 31 C8 CD 6F 34 CD E2 : CB
3490 1F 52 65 61 64 69 6E 67 : D9
3498 20 00 CD 38 44 2A 08 30 : CB
34A0 22 70 1F CD A6 1F 38 8B : 06
34A8 C9 CD A3 1F CD 06 20 38 : 83
34B0 F5 C9 CD 0B 35 CD 3B 35 : 08
34B8 CD 61 35 CD 2D 35 28 F5 : AF
34C0 2A D9 45 ED 5B DD 45 AF : 61
34C8 ED 52 3D CD 5D 42 CD 65 : 1A
34D0 44 8C 28 73 29 0D 3A 3A : 15
34D8 00 2A E4 45 E5 ED 5B CE : 4E
34E0 45 B7 ED 52 E5 22 70 1F : D1
34E8 CD 84 35 2A D0 45 2B CD : BD
34F0 84 35 D1 B7 ED 52 23 22 : C5
34F8 72 1F EB 22 6E 1F CD 84 : 7C
---------------------------------
SUM: 47 94 15 36 79 07 34 E1 8317

3500 35 E1 22 E2 45 CD BE 1F : 09
3508 C3 EE 1F FD 21 C0 37 21 : 06
3510 FF FF 22 F2 45 CD 8A 35 : E3
3518 AF 32 EB 45 21 12 30 CB : 3F
3520 F6 21 D7 45 11 DB 45 01 : 65
3528 04 00 ED B0 C9 21 12 30 : CD
3530 CB B6 3A EB 45 3C 32 EB : 44
3538 45 3D C9 CD E2 1F 50 61 : CA
3540 73 73 3A 00 3A EB 45 C6 : 50
3548 31 CD F4 1F CD EE 1F 21 : 0C
3550 CA 45 11 CB 45 01 09 00 : 3A
3558 36 00 ED B0 DD 21 00 00 : D1
3560 C9 ED 5B 08 30 1A B7 CA : E4
3568 2B 30 CD 9C 35 D8 3A EB : F6
3570 45 B7 28 F6 2A F1 45 7C : F6
3578 A5 C4 82 43 3A F3 45 32 : D2
---------------------------------
SUM: 32 31 13 3A BF 94 70 07 8904

3580 F2 45 18 E6 CD BE 1F C3 : A2
```

```
3588 F1 1F 21 00 00 ED 4B D7 : 40
3590 45 03 AF CD 9A 1F ED A1 : 0B
3598 EA 93 35 C9 ED 53 C8 45 : C8
35A0 AF 32 D4 45 1A B7 37 C8 : CA
35A8 2A D0 45 22 E6 45 DD 22 : 8B
35B0 E8 45 2A CA 45 23 22 CA : 75
35B8 45 FE 09 28 20 FE 3B 28 : F5
35C0 61 FE 0D 28 68 FE 20 28 : 42
35C8 14 FE 2E CA 30 36 CD EC : 29
35D0 36 D2 13 43 CD 35 36 1A : B0
35D8 FE 3A 28 01 1B CD A5 3F : 2D
35E0 FE 3B 28 3E FE 0D 28 45 : 17
35E8 FE 2E 28 44 D6 41 38 16 : FD
35F0 FE 1A 3F 38 11 87 4F 06 : 7C
35F8 00 21 E6 37 09 7E 23 66 : 4E
---------------------------------
SUM: BB EB 54 FC 27 C3 2A 90 8683

3600 6F EB CD 14 37 EB DA 16 : 4D
3608 43 CD A6 3F ED 4B 12 30 : 6F
3610 C5 F1 08 CD 3B 37 3A 12 : 49
3618 30 CB 77 20 05 E6 20 C4 : 61
3620 02 43 EB 01 00 00 3E 0D : 7C
3628 ED B1 EB B7 C9 13 B7 C9 : 9C
3630 21 18 39 18 CC 21 D4 45 : 90
3638 36 01 D5 CD F8 36 E5 D9 : C5
3640 E1 D1 3A EB 45 B7 28 17 : 12
3648 CD B2 36 DA 0D 43 ED 43 : 0F
3650 D5 45 2A DD 45 7C B5 28 : BF
3658 56 2B 22 DD 45 D9 C9 CD : 34
3660 B2 36 D2 10 43 ED 4B DB : 20
3668 45 79 CD 9A 1F 23 78 CD : AC
3670 9A 1F 69 60 08 CA 1F 43 : B6
3678 47 1A CD 9A 1F 13 23 10 : 2D
---------------------------------
SUM: 9E 5C 67 00 56 F9 8C 5A 3EB6

3680 F8 3E 0D CD 9A 1F 23 22 : 0E
3688 D5 45 ED 5B D0 45 7B CD : BF
3690 9A 1F 23 7A CD 9A 1F 23 : FF
3698 AF CD 9A 1F 22 DB 45 3A : B1
36A0 12 30 E6 08 D9 C8 D9 ED : 97
36A8 5B E0 45 ED 52 D9 D8 C3 : 33
36B0 1C 43 3A DF 45 A4 67 CD : 95
36B8 94 1F 4F 23 CD 94 1F 2B : D0
36C0 47 B1 37 C8 D5 E5 69 60 : 7A
36C8 08 47 08 CD 94 1F EB BE : 80
36D0 EB 20 12 23 13 10 F4 CD : 24
36D8 94 1F FE 0D 20 07 23 4D : 55
36E0 44 E1 D1 B7 C9 E1 D1 23 : 4B
36E8 23 C3 B2 36 CD 9E 3F D0 : 48
36F0 FE F4 C8 FE F5 C8 37 C9 : 75
36F8 21 00 00 08 AF 08 CD B9 : 66
---------------------------------
SUM: 87 B0 05 70 6C 1C B8 A1 4323

3700 3F C8 4D 44 29 29 09 4F : 42
3708 06 00 09 29 13 08 3C 20 : AF
3710 EC C3 13 43 23 4D 44 1A : D3
3718 B7 37 C8 1A 13 B7 FA 28 : BC
3720 37 BE 20 0C 23 C3 1B 37 : 59
3728 08 7E CD D4 3F 20 07 C9 : 56
3730 1A 13 B7 F2 30 37 69 60 : 06
3738 C3 17 37 E6 7F FE 10 38 : BC
3740 11 D6 30 38 49 01 5B 39 : 2D
3748 6F 26 00 29 09 4E 23 66 : 9E
3750 69 E9 4F 08 28 21 06 00 : F8
3758 21 24 39 09 7E 18 0D 3E : 68
3760 CB 18 05 3E ED 18 01 79 : A5
3768 08 28 0C 2E 08 EC C5 37 : 5A
3770 08 DD 77 00 DD 23 2E 08 : 92
3778 2A D0 45 23 22 D0 45 C9 : 62
---------------------------------
SUM: 13 1E 91 83 6F CC E8 A7 3E84

3780 FE 0F 18 02 FE 0D 3E DD : 4D
3788 28 DE 3E FD 18 DA 08 28 : 63
3790 25 08 C6 20 87 4F 06 00 : EF
3798 21 31 39 09 4E 23 46 18 : 63
37A0 05 4F 08 28 11 2E 08 EC : B7
37A8 C5 37 08 DD 71 00 DD 23 : 52
37B0 DD 70 00 DD 23 2E 08 2A : AD
37B8 D0 45 23 23 22 D0 45 C9 : 5B
37C0 CD 68 37 18 DD 08 F5 D5 : 33
37C8 DD E5 E1 ED 5B 0D 30 E5 : 0D
37D0 B7 ED 52 E1 D1 DA 37 43 : FC
37D8 D5 ED 5B 0F 30 ED 52 D1 : 6C
37E0 D2 37 43 F1 08 C9 16 38 : 5C
37E8 20 38 24 38 3F 38 64 38 : C7
37F0 23 39 23 39 6F 38 74 38 : 0B
37F8 8A 38 23 39 8F 38 23 39 : 41
---------------------------------
SUM: B8 68 FA BD 30 D2 83 CE 3BB8

3800 A0 38 A7 38 C6 38 23 39 : 11
3808 CE 38 FE 38 23 39 23 39 : F4
3810 23 39 23 39 14 39 44 44 : 8D
3818 B0 4E 44 B1 44 43 B2 00 : 2C
3820 49 54 B3 00 50 B4 41 4C : E1
3828 4C B5 43 46 80 50 4C 81 : 27
3830 50 49 49 90 50 44 91 50 49 : E7
3838 52 92 50 44 52 93 00 45 : A2
3840 43 B6 4A 4E 5A B7 45 46 : 2D
3848 42 B8 45 46 57 B9 45 46 : 20
3850 53 BA 42 B8 57 B9 53 BA : 24
3858 41 41 83 49 82 4D BB 45 : 1D
3860 46 4D BB 00 58 BC 58 58 : 12
3868 84 51 55 BD 49 85 00 41 : F6
3870 4C 54 86 00 4E 43 BE 4E : C3
3878 BF 4E 49 94 4E 44 95 4E : 5F
---------------------------------
SUM: 66 84 15 1A 6E 53 5C D1 79CA

3880 49 52 96 4E 44 52 97 4D : F9
3888 C0 00 52 C1 50 C2 00 44 : 29
3890 C3 44 49 52 98 44 49 99 : 60
3898 44 44 52 9A 44 44 9B 00 : 97
38A0 45 47 9C 4F 50 87 00 52 : A0
38A8 C4 55 54 C5 55 54 49 9D : C1
38B0 54 49 52 9E 55 54 44 9F : 19
38B8 54 44 52 A0 52 47 C6 46 : 2F
38C0 46 53 45 54 C7 00 4F 54 : 98
38C8 C8 55 53 48 C9 00 45 54 : 1A
```

```
38D0 CA 4C 41 88 52 41 89 4C : 47
38D8 43 41 8A 52 43 41 8B 4C : BB
38E0 CB 52 CC 4C 43 CD 52 43 : DA
38E8 CE 4C 44 A1 52 44 A2 45 : 7C
38F0 53 CF 53 54 D0 45 54 49 : 7B
38F8 A3 45 54 4E A4 00 42 43 : B3
---------------------------------
SUM: 6B EA 31 52 EA EA 00 4E C0CD

3900 D1 55 42 D2 43 46 8C 45 : 94
3908 54 D3 4C 41 D4 52 41 D5 : F0
3910 52 4C D6 00 4F 52 D7 00 : EC
3918 4C 49 53 54 D8 4E 4C 49 : F7
3920 53 54 D9 00 3F 2F F3 27 : 08
3928 D9 FB 76 00 17 1F 07 0F : 96
3930 37 ED A1 ED A9 ED B1 ED : E6
3938 B9 ED A2 ED AA ED B2 ED : 6B
3940 BA ED B0 ED A0 ED B8 ED : 76
3948 A8 ED 44 ED A3 ED B3 ED : F6
3950 AB ED BB ED 6F ED 67 ED : F0
3958 4D ED 45 BC 3B 7B 3B 25 : 51
3960 3C A6 3D 8A 3B F2 3C 43 : 55
3968 3C 8F 3C BA 3E D5 3E 62 : 74
3970 3F E1 3E 6B 3B 94 3E 69 : 3F
3978 3C 23 3D 1D 3E 93 3C D4 : 9A
---------------------------------
SUM: 2C D3 31 90 C6 90 4E 41 DFFC

3980 3C 15 3A 87 3B 6E 3D 4F : 47
3988 3E 7F 3E EF 3B EC 3B 11 : 5D
3990 3D CE 3D CB 3D C8 3D C5 : 1A
3998 3D A0 3D FC 3D 16 3C 81 : 26
39A0 3B A3 3D BF 3D BC 3D C2 : D2
39A8 3D 7E 3B 35 3E 48 3E 85 : 74
39B0 3A 85 3A A5 3A 68 3A 19 : 93
39B8 43 10 3B 10 3B 10 3B 10 : 34
39C0 3B 10 3B 10 3B 10 3B BF : DB
39C8 3A A2 3A A2 3A 4D 3B 4D : C7
39D0 3B 53 3C 53 3C 53 3C 53 : 3B
39D8 3C 19 43 4C 3C 4C 3C 4C : F4
39E0 3C 4C 3C 4C 3C 4C 3C 4C : 20
39E8 3C 4C 3C 58 3C 58 3C 60 : 4C
39F0 3C 60 3C 79 3C 79 3C 79 : BB
39F8 3C 79 3C 19 43 72 3C 72 : 6D
---------------------------------
SUM: C5 47 C3 6D C4 3F BF 58 8BB2

3A00 3C 72 3C 72 3C 72 3C 72 : B8
3A08 3C 72 3C 72 3C 7E 3C 7E : D0
3A10 3C 86 3C 86 3C CD 3C 41 : 0A
3A18 38 0B FE 0F DC F3 42 01 : 62
3A20 AF 39 C3 48 37 21 E4 41 : 70
3A28 CD B0 41 D0 CD EC 42 C5 : 4E
3A30 CD F3 42 CD 35 41 FE 0C : 4F
3A38 CA 51 3A FE 04 38 17 FE : A4
3A40 0D DA 19 43 FE 0F 3F 38 : C7
3A48 F8 CD 84 37 C1 3E 22 FD : 9E
3A50 E9 3E 32 C1 FD E9 FE 02 : 00
3A58 28 F2 87 87 87 87 F6 43 : 6F
3A60 F5 CD 63 37 F1 C1 FD E9 : F4
3A68 CD 3C 41 38 16 0E F9 FE : 9D
3A70 02 CA 67 37 FE 0D 01 DD : 53
3A78 F9 CA A2 37 FE 0E FE FD : B3
---------------------------------
SUM: D2 16 35 FB 13 DD 8B 7D C427

3A80 CA A2 37 3E 03 87 87 87 : 79
3A88 87 F5 CD FC 42 28 08 CD : 84
3A90 E6 3F F1 F6 01 FD E9 CD : C0
3A98 63 37 CD E2 42 F1 F6 4B : BD
3AA0 FD E9 CD 84 37 CD FC 42 : 79
3AA8 28 07 CD E6 3F 3E 21 FD : 7D
3AB0 E9 CD E2 42 3E 2A FD E9 : 28
3AB8 D6 05 F6 78 C3 68 37 CD : 78
3AC0 3C 41 38 12 FE 05 38 55 : 57
3AC8 FE 0D 38 EC FE 0F 38 4D : C1
3AD0 06 38 C5 C3 3F 3B CD FC : 09
3AD8 42 28 27 FE 49 28 18 FE : 16
3AE0 52 20 0B 13 CD D3 3F 01 : 70
3AE8 ED 5F CA A2 37 1B CD E6 : BD
3AF0 3F 41 0E 3E C3 A2 37 13 : 7B
3AF8 CD D3 3F 20 F0 01 ED 57 : 34
---------------------------------
SUM: 4B 10 B2 08 3A 42 44 4E A016

3B00 18 F2 21 03 42 CD B0 41 : 2E
3B08 D0 CD E2 42 3E 3A FD E9 : 1F
3B10 D6 05 87 87 87 F5 CD 3C : 6E
3B18 41 38 16 FE 05 38 0F FE : D7
3B20 0D 30 18 D6 05 C1 B0 F6 : 97
3B28 40 FE 76 C2 68 37 C3 19 : F1
3B30 43 CD E6 3F F1 F6 06 41 : 63
3B38 C3 A1 37 FE 0F 38 EF CD : 9C
3B40 D1 42 41 F1 FE 30 28 E6 : 81
3B48 F6 46 C3 A1 37 CD D1 42 : B7
3B50 CD F3 42 79 F5 CD 43 41 : C1
3B58 C1 38 05 F6 70 C3 A1 37 : FF
3B60 0E 36 CD A2 37 CD E6 3F : DC
3B68 C3 67 37 21 10 42 CD B0 : 51
3B70 41 D0 C3 19 43 0E 88 21 : E7
3B78 0E 98 21 0E A0 21 0E A8 : 4C
---------------------------------
SUM: C7 50 7E 8A 3D 25 17 D9 7114

3B80 21 0E 90 21 0E 80 21 0E : 9D
3B88 B0 21 0E B8 CD 3C 41 38 : 19
3B90 10 FE 0F 30 18 D6 05 38 : 78
3B98 D9 FE 08 30 D5 B1 C3 68 : C0
3BA0 37 79 F5 CD E6 3F 41 F1 : C9
3BA8 F6 C6 C3 A1 37 CD 80 37 : DB
3BB0 3E 86 B1 CD 68 37 CD D4 : 82
3BB8 42 C3 67 37 CD 35 41 CD : B3
3BC0 F3 42 FE 0C 28 BE FE 02 : 25
3BC8 28 17 FE 0D 38 A4 FE 0F : 33
3BD0 30 A0 F5 CD 84 37 CD 35 : 4F
3BD8 41 C1 B8 20 07 3E 02 18 : 39
3BE0 07 CD 35 41 FE 04 30 8A : 06
3BE8 0E 09 18 22 0E C5 21 0E : 53
3BF0 C1 CD 35 41 FE 03 28 48 : 75
3BF8 38 14 FE 04 28 0F FE 0D : 90
---------------------------------
SUM: 01 24 AE 59 37 6D 3B FA 58AF
```

▶僕は最近読者になった，しがない中学生でーす。僕はものすごくうれしいことがある。それはなんと，X68000を買うことになったのだ。マンモスラッチー！

棚橋　康敏　(14)　愛知県

```
3C00 38 3E FE 0F 30 3A CD 84 : 3E
3C08 37 3E 02 18 01 3D 87 87 : DB
3C10 87 87 B1 C3 68 37 CD 35 : 23
3C18 41 CD F3 42 FE 0C CA 78 : 8F
3C20 3B 0E 42 18 0D CD 35 41 : F3
3C28 CD F3 42 FE 0C CA 75 3B : 86
3C30 0E 4A FE 02 20 0A CD 63 : B2
3C38 37 CD 35 41 FE 04 38 CE : 82
3C40 C3 19 43 CD 35 41 01 D1 : 34
3C48 39 C3 48 37 D6 05 0E 05 : 69
3C50 C3 0F 3C 0E 0B C3 0E 3C : 34
3C58 CD 84 37 3E 2B C3 68 37 : 53
3C60 CD D1 42 41 0E 35 C3 A2 : C9
3C68 37 CD 35 41 01 F3 39 C3 : 6A
3C70 48 37 D6 05 0E 04 C3 0F : 3E
3C78 3C 0E 03 C3 0E 3C CD 84 : AB
--------------------------------
SUM: 98 3A A9 1F 3A 93 AB A6 E680

3C80 37 3E 23 C3 68 37 CD D1 : 98
3C88 42 41 0E 34 C3 A2 37 0E : 6F
3C90 10 18 14 CD 2E 41 0E 18 : 9E
3C98 38 0D FE 04 30 A2 CD F3 : D9
3CA0 42 87 87 87 F6 20 4F CD : 09
3CA8 67 37 CD E6 3F 08 CA 77 : D9
3CB0 37 08 69 60 ED 4B D0 45 : 55
3CB8 03 B7 ED 42 38 0B 25 24 : 75
3CC0 20 0F 7D B7 F2 68 37 18 : 0C
3CC8 08 24 20 05 7D B7 FA 68 : E7
3CD0 37 C3 22 43 CD 2E 41 0E : A9
3CD8 C2 30 1E 21 40 42 CD B0 : 30
3CE0 41 D0 3E C3 CD 68 37 08 : 86
3CE8 CA B6 37 08 CD E6 3F C3 : 74
3CF0 A6 37 CD 2E 41 38 0A 0E : 69
3CF8 C4 CD 0F 3C CD F3 42 18 : F6
--------------------------------
SUM: 3A D1 1B 2C 07 42 EE C6 9633

3D00 05 3E CD CD 68 37 08 CA : 4E
3D08 B6 37 08 CD E6 3F C3 A6 : 50
3D10 37 CD D3 3F 3E C9 CA 68 : 4F
3D18 37 CD 2E 41 38 29 0E C0 : A2
3D20 C3 0F 3C CD 3C 41 38 1F : AF
3D28 CD F3 42 D6 05 38 18 FE : 2B
3D30 06 28 14 FE 07 28 13 30 : B2
3D38 0E 47 CD 63 37 78 0E 40 : 82
3D40 CD 0F 3C CD 5A 3D C8 C3 : 07
3D48 19 43 21 67 3D CD B0 41 : DF
3D50 D0 CD EC 42 41 0E DB C3 : B8
3D58 A2 37 21 67 3D 06 03 1A : C1
3D60 BE C0 13 23 10 F9 C9 28 : AE
3D68 43 29 00 ED 78 00 D5 CD : 73
3D70 5A 3D 28 16 D1 CD EC 42 : A1
3D78 CD F3 42 CD 43 41 38 1D : A8
--------------------------------
SUM: 4D EF 1C EE F4 A6 2C 5A 9429

3D80 FE 07 20 19 41 0E D3 C3 : 23
3D88 A2 37 C1 CD 63 37 CD F3 : C1
3D90 42 CD 43 41 38 07 FE 06 : D6
3D98 0E 41 C2 0F 3C C3 19 43 : 7B
3DA0 0E 80 21 0E C0 21 0E 40 : EC
3DA8 CD 9D 3F 38 F0 FE 08 30 : 07
3DB0 EC 87 87 87 B1 4F 13 CD : 61
3DB8 F3 42 18 14 0E 28 21 0E : C6
3DC0 20 21 0E 38 21 0E 08 21 : DF
3DC8 0E 00 21 0E 18 21 0E 10 : 94
3DD0 CD 35 41 FE 0F 30 11 D6 : 67
3DD8 05 38 C2 FE 08 30 BE 47 : 3A
3DE0 CD 5F 37 78 B1 C3 68 37 : EE
3DE8 C5 CD 80 37 CD 5F 37 CD : 79
3DF0 D4 42 CD 67 37 C1 79 F6 : B1
3DF8 06 C3 68 37 CD E6 3F 05 : 5F
--------------------------------
SUM: 16 F1 03 A6 59 FD 3D 97 C0CA

3E00 04 20 0F 79 FE 08 38 0D : F7
3E08 4F E6 F8 B9 20 04 FE 40 : 48
3E10 38 06 C3 19 43 87 87 87 : F2
3E18 F6 C7 C3 68 37 1A 13 D6 : 22
3E20 30 28 0A 38 ED FE 03 30 : B8
3E28 E9 3C 87 87 87 F6 46 47 : 3D
3E30 0E ED C3 A2 37 3A F1 45 : 07
3E38 32 F3 45 CD D3 3F C8 CD : DE
3E40 E6 3F 0B ED 43 CA 45 C9 : 38
3E48 21 00 00 22 F2 45 C9 CD : 10
3E50 E6 3F 2A D0 45 ED 43 CC : 60
3E58 45 ED 43 D0 45 ED 43 E6 : A0
3E60 45 37 ED 42 D2 34 43 2A : 1E
3E68 CE 45 09 22 E8 45 E5 DD : 2D
3E70 E1 3A D2 45 B7 C0 22 E4 : AF
3E78 45 3E FF 32 D2 45 C9 2A : BE
--------------------------------
SUM: 45 76 65 6B 18 81 79 90 57D9

3E80 D2 45 7C B5 C2 31 43 3E : BC
3E88 FF 32 D3 45 CD E6 3F ED : 28
3E90 43 CE 45 C9 21 D4 45 35 : 8E
3E98 34 CA B0 3E 34 CD E6 3F : 12
3EA0 ED 43 E6 45 2A D5 45 79 : 18
3EA8 CD 9A 1F 23 78 C3 9A 1F : 9D
3EB0 ED 5B C8 45 C3 1F 43 CD : 47
3EB8 A5 3F 1A CD E0 3F 20 05 : 0F
3EC0 CD E8 3E 18 07 CD E6 3F : 04
3EC8 79 CD 68 37 CD B0 3F 28 : C9
3ED0 E6 C9 CD A5 3F CD E6 3F : 52
3ED8 CD A2 37 CD B0 3F 28 F2 : 7C
3EE0 C9 1A CD E0 3F C2 2E 43 : 02
3EE8 08 38 29 08 13 1A FE 0D : A9
3EF0 CA 2E 43 FE 5E 28 18 CD : A4
3EF8 E0 3F 20 0E 13 4F 1A CD : 96
--------------------------------
SUM: 08 65 2E 30 AF 8A 80 8B A347

3F00 E0 3F 28 04 CD BA 3F C8 : D9
3F08 1B 79 CD 68 37 18 DD CD : C2
3F10 51 3F 18 F6 08 13 1A FE : D1
3F18 0D CA 2E 43 FE 5E 28 2C : F8
3F20 CD 8F 3F 38 0F CD 68 37 : 4E
3F28 13 1A FE 0D CA 2E 43 CD : 40
3F30 68 37 18 E1 CD E0 3F 20 : A4
3F38 0E 13 4F 1A CD E0 3F 28 : 9E
3F40 04 CD BA 3F C8 1B 79 CD : F3
3F48 68 37 18 C9 CD 51 3F 18 : F5
3F50 F6 13 1A FE 0D CA 2E 43 : 69
3F58 D6 40 38 03 FE 1C D8 C6 : 09
3F60 40 C9 CD E6 3F 78 B1 CA : EE
3F68 25 43 DD 09 2A D0 45 09 : 96
3F70 22 D0 45 ED 4B E8 45 DD : 79
3F78 22 E8 45 08 C8 F0 DD E5 : D1
--------------------------------
SUM: 90 CF 37 D2 99 70 5D 8E E21B

3F80 E1 AF ED 42 E5 C5 E1 C1 : 0B
3F88 77 ED A1 EA 88 3F C9 FE : 7D
3F90 81 D8 FE A0 3F D0 FE E0 : E4
3F98 D8 FE F0 3F C9 1A D6 30 : EE
3FA0 D8 FE 0A 3F C9 13 1A FE : 13
3FA8 09 28 FA FE 20 28 F6 C9 : 30
3FB0 CD A6 3F FE 2C C8 FE 3A : DC
3FB8 C9 1A D9 21 C5 3F 01 0E : F0
3FC0 00 ED B1 D9 C9 09 20 0D : 76
3FC8 3B 3A 29 2C 2B 2D 2A 2F : 7B
3FD0 25 22 27 1A FE 09 C8 FE : 55
3FD8 0D C8 FE 3B C8 FE 20 C9 : BD
3FE0 FE 27 C8 FE 22 C9 CD 0C : AF
3FE8 40 CD A6 3F FE 2B 20 08 : 43
3FF0 E5 CD 0B 40 C1 09 18 F1 : D0
3FF8 FE 2D 20 0C E5 CD 0B 40 : 54
--------------------------------
SUM: B6 57 30 4A CF 37 CF 26 942C

4000 4D 44 E1 B7 ED 42 18 E1 : 51
4008 4D 44 C9 13 CD 66 40 CD : AD
4010 A6 3F FE 2A 20 0A E5 CD : E9
4018 65 40 C1 CD 86 42 18 EF : 02
4020 FE 2F 20 0C E5 CD 65 40 : B0
4028 4D 44 E1 CD 99 42 18 DF : 11
4030 FE 25 C0 E5 CD 65 40 4D : 87
4038 44 E1 D5 CD B0 42 D1 18 : A2
4040 CE CD 65 40 7C 2F 67 7D : CF
4048 2F 6F 23 C9 4F 13 1A FE : 04
4050 0D 28 0F FE 5E 20 03 CD : 90
4058 51 3F 6F 26 00 13 1A 13 : 65
4060 B9 C8 C3 2E 43 13 CD A6 : 3B
4068 3F FE 2D 28 D4 FE 24 28 : B0
4070 48 FE 25 28 61 CD 9E 3F : 9E
4078 30 70 C6 30 CD E0 3F 28 : AA
--------------------------------
SUM: FD 57 E0 27 C9 DD 4F 7E 4921

4080 CB 21 C7 3F 01 0C 00 ED : EC
4088 B1 CA 25 43 08 F5 D5 CD : 82
4090 F8 36 E5 D9 E1 D1 CD B2 : 1D
4098 36 30 0D 3A EB 45 B7 C2 : 56
40A0 0D 43 21 00 00 E5 18 0C : 7A
40A8 69 60 CD 94 1F 4F 23 CD : 88
40B0 94 1F 47 C5 D9 E1 F1 08 : 72
40B8 C9 13 1A CD B8 1F 30 04 : CE
40C0 2A D0 45 C9 6F 26 00 13 : B0
40C8 1A CD B8 1F D8 29 29 29 : 11
40D0 29 B5 6F 13 18 F2 21 00 : 8B
40D8 00 13 1A FE 5F 28 FA D6 : 82
40E0 30 D8 FE 02 D0 0F ED 6A : 3E
40E8 18 EF D5 4F 13 CD B9 3F : 03
40F0 20 FA 1B 1A D1 FE 48 20 : 86
40F8 0C 79 CD C4 40 1A FE 48 : B6
--------------------------------
SUM: 5E C5 6E E3 37 A8 E5 36 26BF

4100 C2 25 43 13 C9 FE 42 20 : 66
4108 11 79 FE 02 D2 25 43 6F : 33
4110 26 00 CD D9 40 1A FE 42 : 66
4118 18 E6 69 26 00 13 CD 9D : 0A
4120 3F D8 4D 44 29 29 09 29 : 2C
4128 4F 06 00 09 18 EF 21 9B : 21
4130 41 06 08 18 13 CD 3C 41 : C4
4138 D0 C3 19 43 21 6A 41 06 : C1
4140 11 18 05 21 79 41 06 08 : 17
4148 C5 48 D5 1A BE 20 04 13 : F1
4150 23 18 F8 34 35 20 0A CD : 93
4158 BA 3F 20 05 E1 79 90 C1 : C9
4160 C9 CD 56 42 D1 10 E3 C1 : B3
4168 37 C9 42 43 00 44 45 00 : 0E
4170 48 4C 00 53 50 00 41 46 : BE
4178 00 42 00 43 00 44 00 45 : 0E
--------------------------------
SUM: AB 06 6F 4B BE 31 04 6E EA8B

4180 00 48 00 4C 00 28 48 4C : 50
4188 29 00 41 00 49 58 00 49 : 54
4190 59 00 28 49 58 00 28 49 : 93
4198 59 00 00 4E 5A 00 5A 00 : 5B
41A0 4E 43 00 43 00 50 4F 00 : 73
41A8 50 45 00 50 00 4D 00 00 : 32
41B0 34 35 37 C8 4B 42 1A BE : CD
41B8 20 05 13 23 C3 B6 41 34 : 49
41C0 35 23 20 05 CD D4 3F 28 : 85
41C8 0A CD 56 42 23 23 59 50 : 5E
41D0 C3 B0 41 4E 23 46 78 B7 : 9A
41D8 20 05 CD 67 37 B7 C9 CD : DD
41E0 A2 37 B7 C9 28 44 45 29 : 33
41E8 2C 41 00 12 00 28 42 43 : 2C
41F0 29 2C 41 00 02 00 52 2C : 16
41F8 41 00 ED 4F 49 2C 41 00 : 33
--------------------------------
SUM: 27 53 1C 87 C6 A1 67 64 265B

4200 ED 47 00 44 45 29 00 1A : 00
4208 00 42 43 29 00 0A 00 00 : B8
4210 44 45 2C 48 4C 00 EB 00 : 34
4218 41 46 2C 41 46 27 00 08 : 69
4220 00 28 53 50 29 2C 48 4C : B4
4228 00 E3 00 28 53 50 29 2C : 03
4230 49 58 00 DD E3 28 53 50 : 2C
4238 29 2C 49 59 00 FD E3 00 : D7
4240 28 48 4C 29 00 E9 00 28 : F6
4248 49 58 29 00 DD E9 28 49 : 01
4250 59 29 00 FD E9 00 AF BE : D5
4258 23 C8 C3 57 42 D5 F5 01 : 12
4260 0A 05 11 C5 45 CD A0 42 : D9
4268 C6 30 1B 12 10 F7 6B 62 : F7
4270 01 20 04 3E 30 BE 20 04 : 75
4278 71 23 10 F9 F1 B7 28 01 : 6E
--------------------------------
SUM: 13 AC AF 2F B4 DB B1 C3 7860

4280 EB CD E5 1F D1 C9 D5 EB : 16
4288 21 00 00 3E 10 29 EB 29 : AC
4290 EB 30 01 09 3D 20 F6 D1 : 49
4298 C9 D5 CD B0 42 EB D1 C9 : E2
42A0 C5 AF 06 10 29 17 2C 91 : 87
42A8 30 02 2D 81 10 F6 C1 C9 : 70
42B0 78 B1 28 71 EB 21 00 00 : CE
42B8 3E 10 EB 29 EB ED 6A 1C : C0
42C0 ED 42 30 02 09 1D 3D 20 : E4
42C8 F1 C9 CD E2 42 AF 91 4F : 3A
42D0 C9 CD 80 37 1A 13 D6 29 : 79
42D8 4F C8 FE 04 28 EC FE 02 : 2D
42E0 20 37 CD E6 3F 1A FE 29 : 8A
42E8 20 3E 13 C9 CD FC 42 20 : 65
42F0 28 18 EF 47 1A FE 2C 78 : 32
42F8 20 31 13 C9 1A FE 28 C0 : 2D
--------------------------------
SUM: E9 A2 56 1F 3C F5 14 3F BD04

4300 13 C9 CD A6 3F FE 0D C8 : 61
4308 FE 3B C8 AF 01 3E 01 01 : F1
4310 3E 02 01 3E 03 01 3E 04 : C5
4318 01 3E 05 01 3E 06 01 3E : C8
4320 07 01 3E 08 01 3E 09 01 : 97
4328 3E 0A 01 3E 0B 01 3E 0C : DD
4330 01 3E 0D 01 3E 0E 01 3E : D8
4338 0F ED 53 06 30 87 21 20 : 4D
4340 45 4F 06 00 09 5E 23 56 : 7A
4348 3A F4 45 B7 20 29 CD EE : 2E
4350 1F CD 72 44 CD E2 1F 20 : 90
4358 45 72 72 6F 72 0D 00 3A : 51
4360 EB 45 3C 28 0F CD B6 43 : 69
4368 3A FA 45 47 CD DF 1F 3E : B9
4370 5E CD F4 1F C3 2B 30 ED : 49
4378 7B C6 45 21 2B 30 E5 C3 : AA
--------------------------------
SUM: 86 BE 23 FA 2D 94 AF 45 2982

4380 1D 46 D5 CD 43 44 2A E6 : 9C
4388 45 CD BE 1F 3A D4 45 FE : 40
4390 02 20 09 CD F1 1F 3E 50 : 96
4398 CD F4 1F B7 C4 10 44 06 : B5
43A0 10 CD DF 1F CD B6 43 CD : 6E
43A8 FA 43 20 FB CD D6 1F CD : E7
43B0 C7 1F 7A 33 D1 C9 D5 CD : CF
43B8 F1 1F 2A CA 45 AF CD 5D : 22
43C0 42 CD F1 1F CD F1 1F ED : E9
43C8 5B C8 45 2A 06 30 B7 ED : 6C
43D0 52 20 07 CD 18 20 7D 32 : 2D
43D8 EA 45 1A 13 FE 0D 28 17 : A6
43E0 FE 09 20 0E CD 18 20 3E : 78
43E8 08 85 E6 F8 47 CD B5 32 : 66
43F0 18 D9 CD F4 1F 18 D4 D1 : 8E
43F8 18 13 CD 2A 44 C8 E5 ED : 00
--------------------------------
SUM: 02 E9 55 D4 42 5E FE 4F 6DD6

4400 4B CE 45 B7 ED 42 CD BE : CF
4408 1F E1 CD 14 44 C3 EE 1F : F5
4410 CD 2A 44 C8 06 04 7B B2 : 3A
4418 C8 1B 7E 23 22 E8 45 CD : A0
4420 F1 1F CD C1 1F 10 EF AF : 6B
4428 3D C9 DD E5 E1 ED 5B E8 : D9
4430 45 D5 B7 ED 52 EB E1 C9 : A5
4438 3A 5D 1F CD 51 31 CD 9D : 6F
4440 1F 18 CA 3A 11 30 B7 C8 : FB
4448 CD D9 1F AF CD DC 1F D0 : 0C
4450 AF 32 11 30 CD D6 1F 3E : 22
4458 02 C3 33 20 FE 51 C8 FE : 2D
4460 53 C8 FE 54 C9 E3 7E 23 : BA
4468 B7 28 05 CD 81 44 18 F6 : 84
4470 E3 C9 F5 D5 1A 13 B7 28 : 82
4478 05 CD 81 44 18 F6 D1 F1 : 67
--------------------------------
SUM: 3B 7A FA 89 21 6D 4E 5F 8086

4480 C9 B7 F2 F4 1F D5 E5 87 : C6
4488 6F 26 00 11 98 44 19 5E : F9
4490 23 56 CD E5 1F E1 D1 C9 : C5
4498 51 45 B2 44 BB 44 EF 44 : BE
44A0 B9 45 C4 44 05 45 CB 44 : 5F
44A8 D3 44 0A 32 E9 44 95 34 : 49
44B0 50 45 4D 69 73 73 69 6E : 08
44B8 67 20 00 49 6C 6C 65 67 : 74
44C0 61 6C 20 00 20 54 61 62 : 24
44C8 6C 65 00 53 6F 75 72 63 : DD
44D0 65 20 00 4F 62 6A 65 63 : 68
44D8 74 20 00 52 65 77 69 6E : 99
44E0 64 20 00 4B 61 6E 6A 69 : 71
44E8 00 53 61 76 65 00 84 20 : 33
44F0 43 68 65 63 6B 00 80 85 : E3
44F8 83 00 46 72 61 63 74 69 : DC
--------------------------------
SUM: BF 52 B8 E0 46 21 6F 4C E0E8

4500 6F 6E 61 6C 20 41 73 73 : F1
4508 65 6D 62 6C 65 00 45 4F : 99
4510 4C 83 00 00 43 6C 65 61 : 44
4518 72 20 53 70 61 63 65 00 : 7E
4520 40 45 47 45 57 45 64 45 : 56
4528 67 45 70 45 79 45 84 45 : E8
4530 87 45 8F 45 9B 45 A0 45 : 65
4538 A5 45 AC 45 B4 45 B9 45 : D2
4540 53 79 6E 74 61 78 00 55 : DC
4548 6E 64 65 66 69 6E 65 64 : 3D
4550 20 4C 61 62 65 6C 00 52 : 52
4558 65 64 65 66 69 6E 69 74 : 48
4560 69 6F 6E 00 82 80 00 82 : CA
4568 4F 70 65 63 6F 64 65 00 : BF
4570 82 4F 70 65 72 61 6E 64 : 4B
4578 00 54 6F 6F 20 4D 61 6E : 6E
--------------------------------
SUM: E5 A1 53 35 63 76 C5 0A 1528

4580 79 8C 73 00 81 80 00 54 : CD
```

THE SENTINEL

▶というわけで，沙羅曼蛇は面白いが，簡単すぎる。ドラスピは面白くてムズイが，お勧めはやはりサンダーフォースIIであろう。私はヒーコラいいながら，5面の横スクロールに一度行ったきりだ。それにしても3面，4面のボスは買いである。もう一度5面に行きたい。

西野　裕　(21)　京都府

```
4588 6F 6F 20 46 61 72 00 82 : 99
4590 45 78 70 72 65 73 73 69 : 53
4598 6F 6E 00 81 5B 29 5D 00 : 3F
45A0 81 5B 2C 5D 00 81 51 75 : AC
45A8 6F 74 65 00 82 4F 46 46 : A5
45B0 53 45 54 00 82 4F 52 47 : 56
45B8 00 4D 65 6D 6F 72 79 00 : 79
45C0 00 00 00 00 00 00 00 00 : 00
45C8 00 00 00 00 00 00 00 00 : 00
45D0 00 00 00 00 00 00 00 00 : 00
45D8 00 00 00 00 00 00 00 00 : 00
45E0 00 00 00 00 00 00 00 00 : 00
45E8 00 00 00 00 00 00 00 00 : 00
45F0 00 00 00 00 00 00 00 00 : 00
45F8 00 00 00 00 00 00 00 00 : 00
----------------------------------
SUM: DF 42 4D 03 15 1F 32 41 1B69

4600 00 00 00 00 00 00 00 00 : 00
4608 00 00 00 00 00 00 00 00 : 00
4610 00 00 00 00 00 00 00 C3 : C3
4618 48 46 C3 64 46 2A 06 30 : 5B
4620 D5 CD 60 4B D1 CD F4 4F : 2E
4628 CD 0A 30 3E 3F CD F4 1F : 64
4630 CD C4 1F 21 91 46 E5 C3 : 50
4638 D3 4B 00 00 00 00 00 00 : 1E
4640 00 00 00 00 00 00 00 00 : 00
4648 DD 2A 08 30 DD 6E 00 DD : 67
4650 66 01 DD 7E 02 22 45 46 : 71
4658 32 47 46 DD 36 00 0D DD : BC
4660 36 01 00 C9 3A 5B 1F 32 : E6
4668 3C 46 AF 2A 08 30 4F 47 : 29
4670 ED B1 77 22 3A 46 DD 2A : BE
4678 08 30 DD 22 3D 46 DD 36 : CD
----------------------------------
SUM: 66 C6 A0 D0 B5 B1 4D FD 8FC1

4680 FF 0D DD 77 FE 21 01 00 : 80
4688 22 3F 46 22 CE 4C CD 0C : BC
4690 49 21 91 46 E5 CD F4 4F : 36
4698 3E 3E CD F4 1F CD 21 20 : 6A
46A0 E6 DF FE 51 20 02 E1 C9 : E0
46A8 21 C5 46 06 0C CD B2 46 : 03
46B0 D8 E9 BE 20 09 23 F5 7E : 3E
46B8 23 66 6F F1 B7 C9 23 23 : AF
46C0 23 10 EF 37 C9 47 E5 48 : 96
46C8 56 15 49 5A 53 49 45 CD : BC
46D0 4B 4C 0C 49 4E AD 48 52 : 81
46D8 CE 48 46 C1 49 54 B2 48 : B4
46E0 53 E0 47 43 0B 47 4D E9 : 45
46E8 46 CD F4 4F CD F1 1F 2A : 5D
46F0 08 30 CD BE 1F 3E 2D F5 : 42
46F8 CD F4 1F CD 33 4E CD BE : B9
----------------------------------
SUM: AA 28 A3 F3 99 17 18 A0 D101

4700 1F F1 CD F4 1F 2A 3A 46 : 9A
4708 C3 BE 1F CD 2F 48 FE 1B : FD
4710 C8 11 C2 47 CD A9 4F CD : 74
4718 78 48 C8 1A B7 C8 21 C9 : 0B
4720 47 CD 5A 48 21 96 48 CD : 82
4728 B5 47 E5 21 C9 47 CD B5 : 94
4730 47 D1 B7 ED 52 22 DE 47 : 55
4738 CD A4 47 30 0B CD 21 20 : 01
4740 FE 0D 28 0D FE 1B 20 F0 : 69
4748 CD C4 1F CD 0C 49 C3 D3 : 68
4750 4B 2A DE 47 7C B7 F4 20 : E1
4758 4E CA A3 4A 3E C9 32 62 : A0
4760 4C 32 7C 4C 2A DE 47 7D : 12
4768 B4 28 1E E5 7C E6 80 20 : E1
4770 0D 2A AB 48 2B 22 AC 4E : 71
4778 E1 CD 59 4C 18 0B 2A AB : 4B
----------------------------------
SUM: 84 A7 19 D8 C6 84 62 BB B50B

4780 48 2B 22 AC 4E E1 CD 64 : A1
4788 4C 3E 18 32 62 4C 3E 2A : EA
4790 32 7C 4C 21 C9 47 ED 5B : 73
4798 AB 48 1B 7E B7 28 99 23 : 27
47A0 12 13 18 F7 CD 33 4E CD : 4F
47A8 FD 47 FE 1B C8 2A A7 4C : 42
47B0 CD 1E 20 37 C9 C5 4D 44 : 61
47B8 7E 23 B7 20 FB B7 ED 42 : 59
47C0 C1 C9 52 65 70 6C 63 65 : E5
47C8 3A 20 20 20 20 20 20 20 : 1A
47D0 20 20 20 20 20 20 20 20 : 00
47D8 20 20 20 20 20 00 00 00 : A0
47E0 CD 2F 48 FE 1B C8 CD A4 : 96
47E8 47 D2 C4 1F CD 21 20 FE : 08
47F0 0D 28 F3 FE 1B 20 F5 CD : 23
47F8 C4 1F C3 D3 4B 2A 44 4E : 80
----------------------------------
SUM: EB 39 02 99 A7 54 89 0D FF85

4800 ED 5B AB 48 B7 ED 52 7D : AE
4808 B4 3E 1B C8 4D 44 EB 11 : 62
4810 96 48 1A ED B1 3E 1B C0 : AF
4818 22 AB 48 2B 23 13 1A B7 : 47
4820 28 05 BE 28 F7 18 D6 ED : E5
4828 5B AB 48 1B C3 64 4B CD : A8
4830 33 4E CD F4 4F 11 82 48 : 6C
4838 CD E5 1F CD 21 20 E6 DF : A4
4840 FE 42 2A 3D 46 20 03 2A : 3A
4848 08 30 22 AB 48 11 8F 48 : 35
4850 CD A9 4F CD 78 48 C8 21 : 3B
4858 96 48 06 14 1A 13 77 B7 : 53
4860 C8 FE 5E 28 04 23 10 F4 : 77
4868 C9 1A 13 FE 5E 28 04 E6 : 64
4870 DF D6 40 77 23 10 E5 C9 : 4D
4878 D5 ED 5B 76 1F 1A D1 FE : 9B
----------------------------------
SUM: 8A AD C7 08 C6 30 96 D1 1CE7

4880 1B C9 42 65 67 69 6E 2C : F5
4888 20 48 65 72 65 20 00 53 : 17
4890 65 61 72 63 68 3A 20 20 : 7D
4898 20 20 20 20 20 20 20 20 : 00
48A0 20 20 20 20 20 20 20 20 : 00
48A8 20 20 00 00 00 CD 48 46 : 9B
48B0 18 2F 11 C4 48 CD A9 4F : 29
48B8 CD B2 1F D8 22 08 30 CD : 9D
48C0 48 46 18 1D 54 65 78 74 : 68
48C8 20 41 64 72 3A 00 DD 2A : 78
48D0 08 30 2A 45 46 DD 75 00 : 3F
48D8 DD 74 01 3A 47 46 DD 77 : 6D
48E0 02 E1 C3 64 46 11 FF 48 : A8
48E8 CD A9 4F CD 78 48 C8 CD : E7
48F0 85 4F CD 47 4F ED 53 3F : B6
48F8 46 22 3D 46 C3 0C 49 47 : 4A
----------------------------------
SUM: CC D9 4C E2 C9 7F F9 F1 189E

4900 6F 74 6F 3A 00 FE 30 D8 : 92
4908 FE 3A 3F C9 ED 5B 3D 46 : 0B
4910 2A 3F 46 18 11 ED 5B 41 : 61
4918 46 1A B7 C8 ED 53 3D 46 : A2
4920 2A 43 46 22 3F 46 3E 0C : A4
4928 CD F4 1F 3A 3C 46 3D F5 : CE
4930 CD 0D 4F CD CD 4E 30 13 : 54
4938 F1 91 38 11 ED 53 41 46 : 92
4940 D5 11 00 2E CD E5 1F D1 : B6
4948 23 18 E4 F1 1B 22 43 46 : D6
4950 C9 00 00 3A 3C 46 3D 32 : F4
4958 C0 49 67 ED 5B 3F 46 1B : 58
4960 7B B2 C8 7C 2A 3D 46 22 : 40
4968 3D 46 2B 2B F5 CD 8C 49 : 70
4970 30 09 F1 1B 91 30 F0 81 : 77
4978 13 18 01 F1 32 C0 49 13 : 6B
----------------------------------
SUM: 0E 67 C7 16 81 4C E1 62 1961

4980 ED 53 3F 46 EB ED 5B 3D : 35
4988 46 C3 26 49 7E B7 20 03 : D0
4990 23 23 C9 3E 0D 01 00 00 : 5B
4998 ED B9 23 23 E5 0E 00 7E : 5D
49A0 0C 23 FE 0D 28 0C FE 09 : 75
49A8 20 F5 79 E6 F8 C6 08 4F : 89
49B0 18 ED 69 26 00 01 04 00 : 99
49B8 09 CD 04 4F 37 0C E1 C9 : 16
49C0 00 11 C4 4A CD E5 1F CD : BD
49C8 21 20 E6 DF FE 4C 28 14 : 8C
49D0 FE 53 28 10 FE 43 28 0C : FE
49D8 FE 47 28 08 FE 44 CA 01 : 82
49E0 4B C3 0C 49 F5 11 E8 4A : 9B
49E8 CD A9 4F CD 78 48 20 02 : 74
49F0 F1 C9 D5 21 ED 4A EB 01 : D3
49F8 13 00 ED B0 D1 3E 04 CD : 90
----------------------------------
SUM: C9 C4 4C 80 A4 2B 96 E7 47A7

4A00 A3 1F F1 FE 53 28 62 FE : 8C
4A08 43 28 5E F5 CD 09 20 DA : 8E
4A10 BA 4A 28 08 CD 9D 1F CD : 8A
4A18 EE 1F 18 F0 F1 FE 4C 28 : 78
4A20 15 CD 33 4E ED 5B 72 1F : 3C
4A28 19 38 78 EB 2A 6A 1F B7 : 1E
4A30 ED 52 30 1B 18 6D 21 01 : 31
4A38 00 22 3F 46 2A 08 30 22 : 2B
4A40 3D 46 22 70 1F CD A6 1F : C6
4A48 F5 38 6F F1 C3 E1 48 CD : 46
4A50 33 4E 23 22 70 1F E5 CD : 07
4A58 A6 1F 38 5E E1 ED 5B 72 : F6
4A60 1F 19 2B 22 3A 46 C3 0C : D4
4A68 49 F5 CD 33 4E F1 FE 43 : BE
4A70 28 06 ED 5B 08 30 18 06 : CC
4A78 23 ED 5B 3A 46 EB B7 ED : 7A
----------------------------------
SUM: 67 15 D5 50 40 12 8D 33 BCA1

4A80 52 23 22 72 1F 21 00 00 : 49
4A88 22 70 1F 21 00 00 22 6E : 62
4A90 1F D5 CD AF 1F 38 23 E1 : CB
4A98 22 70 1F CD AC 1F F5 38 : 76
4AA0 19 F1 C9 CD C4 1F CD F4 : 44
4AA8 4F 11 AF 4A C3 E5 1F 20 : 40
4AB0 4E 4F 20 4D 45 4D 4F 52 : 3D
4AB8 59 00 C1 CD 33 20 CD 21 : 28
4AC0 20 C3 0C 49 4C 6F 61 64 : B8
4AC8 2C 20 53 61 76 65 2C 20 : 27
4AD0 47 65 74 2D 63 75 74 2C : C5
4AD8 20 43 75 74 2D 73 61 76 : C3
4AE0 65 2C 20 44 69 72 20 00 : F0
4AE8 46 49 4C 45 3A 4E 4F 4E : 45
4AF0 41 4D 45 20 20 20 20 20 : 73
4AF8 20 20 20 20 20 20 20 20 : 00
----------------------------------
SUM: 83 96 9F 54 1E A5 53 C2 5957

4B00 00 11 58 4B CD E5 1F CD : 52
4B08 21 20 FE 1B C8 FE 0D 20 : 4D
4B10 03 CD 24 20 32 5D 1F CD : 8F
4B18 EE 1F CD 06 20 11 3B 4B : 97
4B20 CD E5 1F ED 5B 76 1F CD : 7B
4B28 D3 1F 1A FE 1B C8 21 05 : 13
4B30 00 19 11 ED 4A 01 13 00 : 75
4B38 ED B0 C9 48 69 74 20 43 : EE
4B40 52 20 6F 6E 20 66 69 6C : AA
4B48 65 6E 61 6D 65 20 6F 72 : 07
4B50 20 42 52 45 41 4B 0D 00 : 92
4B58 0D 44 72 69 76 65 3A 00 : 41
4B60 ED 5B 06 30 2A 08 30 2B : 0B
4B68 D9 01 01 00 D9 01 00 00 : B5
4B70 3E 0D ED B1 E5 B7 ED 52 : C4
4B78 E1 28 14 30 05 D9 03 D9 : 07
----------------------------------
SUM: 68 8F F6 46 39 D3 38 4E 3CA5

4B80 18 F0 3E 0D 2B 2B BE 2B : 92
4B88 20 FC D9 0B D9 23 23 22 : 41
4B90 AC 4E D9 ED 43 A3 4C D9 : CB
4B98 06 00 E5 B7 ED 52 E1 28 : EA
4BA0 0F 7E FE 09 20 06 78 E6 : 18
4BA8 F8 C6 07 47 04 23 18 EA : 35
4BB0 C5 CD E5 4D C1 78 C6 05 : C8
4BB8 6F 26 00 3A 5C 1F 4F CD : 66
4BC0 33 4F 32 A7 4C 3A A8 4C : D5
4BC8 85 32 A8 4C C9 21 04 00 : 99
4BD0 22 A7 4C 21 D3 4B E5 2A : 63
4BD8 A7 4C CD 1E 20 ED 5B 76 : BC
4BE0 1F CD D3 1F CD 18 20 2E : 11
4BE8 04 22 A7 4C 1A FE 1B 20 : 6C
4BF0 05 CD F4 4F E1 C9 FE 30 : ED
4BF8 D8 FE 3A D0 CD 85 4F 22 : A3
----------------------------------
SUM: A6 9F 5A 4F 12 FA 27 7C 9479

4C00 A3 4C 08 ED 53 A5 4C CD : F5
4C08 47 4F 22 AC 4E CD 46 4E : 13
4C10 08 21 A9 4C 06 0A CD B2 : AD
4C18 46 38 01 E9 08 2A AC 4E : 94
4C20 CD AE 4E E5 CD BC 4E E5 : 6A
4C28 2A AC 4E 7E E1 B7 20 05 : 5F
4C30 23 08 3E 2B 08 08 FE 2B : CD
4C38 20 04 11 0A 00 19 08 D1 : 31
4C40 B7 ED 52 7D B4 28 35 EB : 6F
4C48 2A 41 46 19 22 41 46 EB : 5E
4C50 7C E6 80 20 0F CD 20 4E : 4C
4C58 D8 CD 97 4E ED 5B 3A 46 : 52
4C60 ED B8 18 18 E5 EB 21 00 : C6
4C68 00 B7 ED 52 ED 5B AC 4E : 38
4C70 19 E3 CD 97 4E ED 5B AC : A2
4C78 4E E1 ED B0 2A AC 4E CD : BD
----------------------------------
SUM: FB 6E 2D 1B 81 AA CA 32 CF1E

4C80 8E 4E 28 09 1A B7 28 05 : 0B
4C88 77 23 13 18 F7 36 0D 23 : 22
4C90 08 FE 2E 28 32 FE 2B C0 : 77
4C98 06 0A 3E 0D 77 23 10 FC : 01
4CA0 C3 0C 49 00 00 00 00 00 : 18
4CA8 00 2D 18 4D 2F 18 4D 3E : 64
4CB0 59 4D 3C 90 4D 3D E5 4D : 2E
4CB8 28 08 4D 29 0D 4D 2C D0 : FC
4CC0 4C 5B E2 4C 5D E2 4C 2A : 8A
4CC8 A3 4C 22 CE 4C C9 00 00 : F4
4CD0 2A CE 4C ED 5B A3 4C ED : 68
4CD8 53 CE 4C CD 47 4F EB C3 : 7E
4CE0 EC 4D 08 2A A3 4C 22 3F : BB
4CE8 46 2A AC 4E 22 3D 46 08 : 17
4CF0 FE 5B 20 05 CD 0C 49 18 : B8
4CF8 03 CD 53 49 3A 3C 46 CB : F3
----------------------------------
SUM: F6 E9 54 F6 5A 1E 48 43 F402

4D00 3F 67 2E 04 22 A7 4C C9 : B6
4D08 CD 53 49 18 03 CD 15 49 : AF
4D10 2A A7 4C 25 22 A7 4C C9 : 20
4D18 08 CD 33 4E 23 ED 5B AC : 6D
4D20 4E D5 1A 13 77 23 FE 0D : F5
4D28 20 F8 36 00 22 3A 46 08 : F8
4D30 FE 2F 20 07 1A 13 FE 0D : 8C
4D38 28 FA 1B E1 D5 E5 B7 ED : 7C
4D40 52 CD 97 4E D1 E1 7E B7 : EB
4D48 20 02 2B 03 ED B0 CD 0C : C6
4D50 49 2A A7 4C 25 22 A7 4C : A0
4D58 C9 CD 33 4E 23 ED 5B 3A : BC
4D60 46 D5 EB B7 ED 52 CD 20 : E9
4D68 4E D1 30 06 CD C4 1F C3 : C8
4D70 A3 4A 7D B4 C8 D5 CD 97 : 1F
4D78 4E ED 5B 3A 46 ED B8 E1 : 9C
----------------------------------
SUM: DB C7 10 20 C0 D5 BF 3A 5253

4D80 ED 5B AC 4E 7E B7 28 05 : A4
4D88 23 12 13 18 F7 C3 0C 49 : 6F
4D90 2A CE 4C CD 47 4F EB 2A : BC
4D98 AC 4E E5 D5 B7 ED 52 28 : D2
4DA0 3F 30 0D 2A A3 4C 22 CE : 85
4DA8 4C E1 D1 E5 D5 B7 ED 52 : AE
4DB0 4D 44 CD 20 4E 30 03 E1 : E0
4DB8 E1 C9 CD 33 4E 23 EB ED : F3
4DC0 B0 AF 12 ED 53 3A 46 E1 : 12
4DC8 D1 3E C9 32 4E 4D CD 3C : AE
4DD0 4D 3E CD 32 4E 4D 2A CE : 1D
4DD8 4C CD 47 4F EB C3 EC 4D : 96
4DE0 D1 E1 C3 C4 1F 2A A3 4C : 71
4DE8 ED 5B AC 4E 22 3F 46 ED : D6
4DF0 53 3D 46 3A 3C 46 F5 CB : 52
4DF8 3F F5 32 3C 46 3E C9 32 : 21
----------------------------------
SUM: 09 0D 3E 92 24 90 3E FC 2488

4E00 89 49 CD 53 49 3E C3 32 : 6E
4E08 89 49 F1 47 3A C0 49 ED : 3A
4E10 44 80 3D 67 2E 04 22 A7 : 63
4E18 4C F1 32 3C 46 C3 0C 49 : 09
4E20 E5 D5 ED 5B 3A 46 19 38 : D3
4E28 07 EB 2A 6A 1F B7 ED 52 : 9B
4E30 D1 E1 C9 C5 2A 3A 46 2B : 15
4E38 01 00 00 AF ED B9 22 44 : BC
4E40 4E 23 C1 C9 00 00 CD 8E : 56
4E48 4E C8 08 B7 08 0E 00 6B : 56
4E50 62 1A 77 B7 C8 FE 22 28 : BA
4E58 04 FE 27 20 03 08 37 08 : 93
4E60 FE 20 20 04 CD 6D 4E 08 : D2
4E68 23 13 0C 18 E4 08 D8 08 : 26
4E70 E5 79 E6 07 ED 44 C6 08 : 4A
4E78 47 1A 77 FE 20 20 0C 13 : 35
----------------------------------
SUM: AF 6D FD EE F8 A2 C6 5C E382

4E80 23 0C 10 F5 1B 0D E1 36 : 73
4E88 09 08 C9 F1 08 C9 ED 5B : E4
4E90 A5 4C 1B 1A 13 B7 C9 ED : A6
4E98 5B 3A 46 19 22 3A 46 2A : C0
4EA0 AC 4E D5 EB B7 ED 52 23 : D3
4EA8 4D 44 E1 C9 00 00 E5 7E : 9E
4EB0 23 B7 28 04 FE 0D 20 F7 : 28
4EB8 D1 ED 52 C9 CD 8E 4E 6B : ED
4EC0 62 23 28 06 2B 7E 23 B7 : 36
4EC8 20 FB ED 52 C9 E5 21 05 : 2E
4ED0 2E 0E 00 71 1A B7 28 29 : CF
4ED8 13 FE 09 20 12 79 E6 07 : B2
4EE0 ED 44 C6 08 3D 47 3E 20 : E1
4EE8 28 05 77 0C 23 10 FB 77 : 55
4EF0 23 0C FE 0D 20 DE 36 00 : 6E
4EF8 01 01 2E ED 42 CD 04 4F : 7F
----------------------------------
SUM: 15 50 F1 91 BC E4 47 7D C29B

4F00 37 0C E1 C9 3A 5C 1F 4F : F1
4F08 CD 33 4F 4D C9 F5 C5 D5 : F4
4F10 E5 06 04 11 04 2E 3E 20 : 90
4F18 12 1B E5 CD 2E 4F C6 30 : 52
4F20 12 1B 10 F7 13 E1 23 E1 : 2C
4F28 D1 C1 F1 C9 00 00 C5 0E : 1F
4F30 0A 18 01 C5 AF 06 10 29 : D6
```

▶あのー，CRTは1/60秒で走査しているんだったら，1/60×５秒くらいでCRTに「合格」とか，「落ちるな」とか書いて数分おきにでも表示させれば，刷り込まれるのでしょうか。僕はぜひ試してみたい。

中村　賢治　(19)　兵庫県

```
4F38 8F B9 30 05 10 F9 C3 45 : 8E
4F40 4F 91 23 10 F2 C1 C9 7D : 0C
4F48 B4 20 03 21 01 00 ED 5B : 41
4F50 3F 46 2B EB B7 ED 52 13 : A4
4F58 D5 D9 D1 D9 38 08 11 01 : AA
4F60 00 2A 08 30 18 07 ED 5B : C9
4F68 3F 46 2A 3D 46 7E B7 C8 : 2F
4F70 CD 7D 4F C8 7E 23 FE 0D : 0D
4F78 20 FA 13 18 F0 D5 D9 E1 : C4
---------------------------------
SUM: BA C4 01 C0 B5 E1 37 CE 80A6

4F80 B7 ED 52 D9 C9 C5 21 00 : 7E
4F88 00 3E 04 F5 1A 13 CD 05 : 36
4F90 49 38 13 29 4D 44 29 29 : A0
4F98 09 D6 30 4F 06 00 09 F1 : 5E
4FA0 3D 20 E8 1A 13 F5 C1 C1 : E9
4FA8 C9 CD F4 4F 3E 20 06 23 : 60
4FB0 CD F4 1F 10 FB CD F4 4F : FB
4FB8 CD D3 4F 1A 13 2C FE 3A : 80

4FC0 20 F9 E5 CD 1E 20 ED 5B : 51
4FC8 76 1F CD D3 1F E1 26 00 : 5B
4FD0 19 EB C9 F5 D5 1A 13 B7 : 7B
4FD8 28 17 FE 5E CC F4 1F FE : 78
4FE0 20 30 09 F5 3E 5E CD F4 : AB
4FE8 1F F1 C6 40 CD F4 1F 18 : 0E
4FF0 E4 D1 F1 C9 3A 5B 1F 3D : 60
4FF8 2E 00 67 C3 1E 20 00 0D : A3
---------------------------------
SUM: D1 F9 83 8D D6 06 29 F2 01FF
```

リスト2 アセルブラソースリスト

```
0000               ;##################################################
0000               ;#                                                #
0000               ;#           Z80 Assembler                        #
0000               ;#                                                #
0000               ;#           1988/12/12   written by T.T.         #
0000               ;#                                                #
0000               ;##################################################
0000               ;
0000                       OFFSET  5000H-3000H
3000                       ORG     3000H
3000               ;
4617 P             EDCOLD  EQU     4617H    ;editor cold start
461A P             EDHOT   EQU     461AH    ;editor hot start
461D P             EDERR   EQU     461DH    ;editor error
5000 P             TXTST   EQU     5000H    ;default text address
3000               ;
1FFA P             #HOT    EQU     1FFAH
1FF4 P             #PRINT  EQU     1FF4H
1FF1 P             #PRNTS  EQU     1FF1H
1FEE P             #LTNL   EQU     1FEEH
1FEB P             #NL     EQU     1FEBH
1FE8 P             #MSG    EQU     1FE8H
1FE5 P             #MSX    EQU     1FE5H
1FE2 P             #MPRNT  EQU     1FE2H
1FDF P             #TAB    EQU     1FDFH
1FDC P             #LPRNT  EQU     1FDCH
1FD9 P             #LPTON  EQU     1FD9H
1FD6 P             #LPTOF  EQU     1FD6H
1FD3 P             #GETL   EQU     1FD3H
1FCD P             #BRKEY  EQU     1FCDH
1FC7 P             #PAUSE  EQU     1FC7H
1FC1 P             #PRTHX  EQU     1FC1H
1FBE P             #PRTHL  EQU     1FBEH
1FB8 P             #HEX    EQU     1FB8H
1FB2 P             #HLHEX  EQU     1FB2H
1FAF P             #WOPEN  EQU     1FAFH
1FAC P             #WRD    EQU     1FACH
1FA6 P             #RDD    EQU     1FA6H
1FA3 P             #FILE   EQU     1FA3H
1F9D P             #FPRNT  EQU     1F9DH
1F9A P             #POKE   EQU     1F9AH
1F94 P             #PEEK   EQU     1F94H
2006 P             #DIR    EQU     2006H
2009 P             #ROPEN  EQU     2009H
2018 P             #CSR    EQU     2018H
2021 P             #FLGET  EQU     2021H
2024 P             #RDVSW  EQU     2024H
2033 P             #ERROR  EQU     2033H
3000               ;
1F76 P             #KBFAD  EQU     1F76H
1F74 P             #IBFAD  EQU     1F74H
1F72 P             #SIZE   EQU     1F72H
1F70 P             #DTADR  EQU     1F70H
1F6E P             #EXADR  EQU     1F6EH
1F6A P             #MEMAX  EQU     1F6AH
1F68 P             #WKSIZ  EQU     1F68H
1F5D P             #DSK    EQU     1F5DH
1F5C P             #WIDTH  EQU     1F5CH
3000               ;
3000               ;       constants
3000               ;
0009 P             TAB     EQU     '^I'
000C P             CLR     EQU     '^L'
000D P             CR      EQU     '^M'
001B P             BRK     EQU     '^['
0020 P             SPC     EQU     ' '
003A P             PRMPT   EQU     ':'
3000               ;
3000               ;       dictionary code
3000               ;
0080 P             @LBL    EQU     80H
0081 P             @MIS    EQU     81H
0082 P             @ILL    EQU     82H
0083 P             @CHK    EQU     83H
0084 P             @MEM    EQU     84H
0085 P             @TAB    EQU     85H
0086 P             @ASM    EQU     86H
0087 P             @SOU    EQU     87H
0088 P             @OBJ    EQU     88H
0089 P             @ARE    EQU     89H
008A P             @SAV    EQU     8AH
008B P             @ING    EQU     8BH
008C P             @SLB    EQU     8CH
3000               ;
3000               ;       start of program
3000               ;
3000 CD 37 30      START:  CALL    COLD     ;cold start entry(3000h)
3003 18 22                 JR      HOT      ; hot start entry(3003h)
3005               ;
3005 FF            EDFLG:  DEFB    -1       ;whether editor exists
3006                                        ;       0 ... exists
3006                                        ;       -1 ... not
3006 00 00         ERRPTR: DEFW    0000H    ;error occurred address
3008 00 50         TXAREA: DEFW    TXTST    ;source text start address
300A               ;
300A C3 72 44              JP      @MSX     ;for editor
300D               ;
300D 00 50         MEMLOW: DEFW    TXTST    ;object area start address
300F FF FF         MEMHI:  DEFW    0FFFFH   ;object area end address
3011 00            LPTSW:  DEFB    00H      ;printer on/off switch
3012 00            OPTFLG: DEFB    00H      ;option flags
3013                                        ;       F'=CPEFLMSK=SZ---P-C
3013 41 3A         OBJDEV: DEFB    'A:'     ;device
3015               OBJNAM: DEFS    13       ;name
3022 2E 4F 42 4A           DEFB    '.OBJ'   ;.ext
3026 00                    DEFB    0        ;
3027               ;
3027               ;       hot start
3027               ;
3027 ED 73 C6 45   HOT:    LD      (SPBUF),SP
302B ED 7B C6 45   HOT0:   LD      SP,(SPBUF)
302F CD D6 1F              CALL    #LPTOF
3032 CD C5 30              CALL    CMD
3035 18 F4                 JR      HOT0
3037               ;
3037               ;       cold start
3037               ;
3037 CD 65 44      COLD:   CALL    @MPRNT
303A 0C 2A 20 4E           DEFB    CLR,'* New Z80 ',@ASM,'r *',CR,0
303E 65 77 20 5A
3042 38 30 20 86
3046 72 20 2A 0D
304A 00
304B 3A 05 30              LD      A,(EDFLG)
304E B7                    OR      A
304F C4 17 46              CALL    NZ,EDCOLD
3052 2A 6A 1F              LD      HL,(#MEMAX)
3055 22 0F 30              LD      (MEMHI),HL
3058 CD 2D 32              CALL    XCOM0
305B CD EE 1F              CALL    #LTNL
305E CD FE 31              CALL    PCOM0
3061 CD 65 44              CALL    @MPRNT
3064 0D 80 85 20           DEFB    CR,@LBL,@TAB,' :',0
3068 3A 00
306A 2A 68 1F              LD      HL,(#WKSIZ)
306D CD BE 1F              CALL    #PRTHL
3070 7C                    LD      A,H
3071 11 E0 FF              LD      DE,-0020H
3074 19                    ADD     HL,DE
3075 22 E0 45              LD      (WKMAX),HL
3078 21 00 04              LD      HL,0400H
307B FE 20                 CP      20H                ;#WKSIZ<8K
307D 38 0E                 JR      C,COLD1
307F 26 08                 LD      H,08H
3081 FE 40                 CP      40H                ;#WKSIZ<16K
3083 38 08                 JR      C,COLD1
3085 26 10                 LD      H,10H
3087 FE 80                 CP      80H                ;#WKSIZ<32K
3089 38 02                 JR      C,COLD1
308B 26 20                 LD      H,20H
308D 22 D7 45      COLD1:  LD      (HSHSIZ),HL
3090 CD 65 44              CALL    @MPRNT
3093 48 0D 48 61           DEFB    'H',CR,'Hash',@TAB,'  :',0
3097 73 68 85 20
309B 20 3A 00
309E CD BE 1F              CALL    #PRTHL
30A1 7C                    LD      A,H
30A2 3D                    DEC     A
30A3 32 DF 45              LD      (HSHMSK),A
30A6 CB 3C                 SRL     H
30A8 CB 1D                 RR      L
30AA 2B                    DEC     HL
30AB 22 D9 45              LD      (LBLMAX),HL
30AE CD E2 1F              CALL    #MPRNT
30B1 48 20 28 00           DEFB    'H (',0
30B5 3E FF                 LD      A,-1
30B7 CD 5D 42              CALL    PRDEC
30BA CD 65 44              CALL    @MPRNT
30BD 8C 73 29 0D           DEFB    @SLB,'s)',CR,0
30C1 00
30C2 C3 43 31              JP      VCOM0
30C5               ;
30C5               ;       interpret command
30C5               ;
30C5 CD EB 1F      CMD:    CALL    #NL
30C8 3E 3A                 LD      A,PRMPT
30CA CD F4 1F              CALL    #PRINT
30CD ED 5B 76 1F           LD      DE,(#KBFAD)
30D1 CD D3 1F              CALL    #GETL
30D4 1A                    LD      A,(DE)
30D5 FE 3A                 CP      PRMPT
30D7 C0                    RET     NZ
30D8 CD A5 3F              CALL    SPCUT0
30DB B7                    OR      A
30DC C8                    RET     Z
30DD FE 41                 CP      'A'
30DF 38 02                 JR      C,CMD0
30E1 E6 DF                 AND     0DFH               ;11011111
30E3 08            CMD0:   EX      AF,AF'
30E4 CD A5 3F              CALL    SPCUT0
30E7 08                    EX      AF,AF'
30E8 21 FB 30              LD      HL,CMDTBL
30EB 35            CMD1:   DEC     (HL)
30EC 34                    INC     (HL)
30ED C8                    RET     Z
30EE BE                    CP      (HL)
30EF 23                    INC     HL
30F0 28 04                 JR      Z,CMD2
30F2 23                    INC     HL
30F3 23                    INC     HL
30F4 18 F5                 JR      CMD1
30F6 4E            CMD2:   LD      C,(HL)
30F7 23                    INC     HL
30F8 46                    LD      B,(HL)
30F9 C5                    PUSH    BC
30FA C9                    RET
30FB               ;
30FB 51            CMDTBL: DEFB    'Q'                ;quit
30FC 23 31                 DEFW    QCOM
30FE 41                    DEFB    'A'                ;asm
30FF C0 32                 DEFW    ACOM
3101 4F                    DEFB    'O'                ;label list
3102 6C 32                 DEFW    OCOM
3104 2F                    DEFB    '/'                ;switch
3105 59 31                 DEFW    SWITCH
3107 3F                    DEFB    '?'                ;calc
3108 40 32                 DEFW    ?COM
310A 4A                    DEFB    'J'                ;jump
310B 29 31                 DEFW    JCOM
310D 50                    DEFB    'P'                ;mem
310E EC 31                 DEFW    PCOM
3110 58                    DEFB    'X'                ;text
3111 25 32                 DEFW    XCOM
3113 53                    DEFB    'S'                ;save
3114 F8 33                 DEFW    SCOM
3116 44                    DEFB    'D'                ;dir
3117 A9 34                 DEFW    DCOM
3119 56                    DEFB    'V'                ;dev
311A 36 31                 DEFW    VCOM
311C 23                    DEFB    '#'                ;lpt sw
311D D5 31                 DEFW    #COM
311F 45                    DEFB    'E'                ;edit
3120 2E 31                 DEFW    ECOM
3122 00                    DEFB    0
3123               ;
3123               ;       quit
3123               ;
3123               QCOM:
3123 ED 7B C6 45   EXIT:   LD      SP,(SPBUF)
3127 B7                    OR      A                  ;NC
3128 C9                    RET
3129               ;
3129               ;       jump
3129               ;
3129               JCOM:
3129 CD B2 1F      JUMP:   CALL    #HLHEX
312C D8                    RET     C
312D E9                    JP      (HL)
312E               ;
312E               ;       edit
312E               ;
312E 3A 05 30      ECOM:   LD      A,(EDFLG)
3131 B7                    OR      A
3132 C8                    RET     Z
3133 C3 1A 46              JP      EDHOT
3136               ;
3136               ;       device
3136               ;
3136 1A            VCOM:   LD      A,(DE)
3137 B7                    OR      A
3138 28 09                 JR      Z,VCOM0
313A CD A3 1F              CALL    #FILE
313D 3A 5D 1F              LD      A,(#DSK)
3140 32 13 30              LD      (OBJDEV),A
3143 CD 65 44      VCOM0:  CALL    @MPRNT
3146 88 44 72 69           DEFB    @OBJ,'Drive:',0
314A 76 65 3A 00
314E 3A 13 30              LD      A,(OBJDEV)
```

▶うちの「リュウ」という名前のネコは，最近いたずらにも磨きがかかり，なんと押し入れの戸を勝手に開けてはなかを引っかき回す始末。それでつい最近X68000を買った私は，ネコを部屋に入れなくなってしまった（ちょうどいい机がないので，X68000は押し入れのなかにあったりする）。

田岡　直人（20）神奈川県

```
3151 CD F4 1F      PRDEV:  CALL    #PRINT
3154 3E 3A                 LD      A,':'
3156 C3 F4 1F              JP      #PRINT
3159               ;
3159               ;       option
3159               ;
3159 CD A6 3F      SWITCH: CALL    SPCUT
315C E6 DF                 AND     0DFH            ;11011111
315E 01 08 00              LD      BC,OPTTBE-OPTTBL
3161 21 AB 31              LD      HL,OPTTBL
3164 ED B1                 CPIR
3166 20 22                 JR      NZ,SWT4
3168 0C                    INC     C
3169 41                    LD      B,C
316A AF                    XOR     A
316B 37                    SCF
316C 17            SWT0:   RLA
316D 10 FD                 DJNZ    SWT0
316F 4F                    LD      C,A
3170 13                    INC     DE
3171 21 12 30              LD      HL,OPTFLG
3174 1A                    LD      A,(DE)
3175 FE 2D                 CP      '-'
3177 28 0B                 JR      Z,SWT3
3179 FE 2B                 CP      '+'
317B 28 01                 JR      Z,SWT1
317D 06                    DEFB    06H             ;LD B,n
317E 13            SWT1:   INC     DE
317F 79                    LD      A,C
3180 B6                    OR      (HL)
3181 77            SWT2:   LD      (HL),A
3182 18 D5                 JR      SWITCH
3184 13            SWT3:   INC     DE
3185 79                    LD      A,C
3186 2F                    CPL
3187 A6                    AND     (HL)
3188 18 F7                 JR      SWT2
318A 3A 12 30      SWT4:   LD      A,(OPTFLG)
318D 4F                    LD      C,A
318E 11 E3 44              LD      DE,SWMES
3191 06 08                 LD      B,8
3193 1A            SWT5:   LD      A,(DE)
3194 B7                    OR      A
3195 28 0A                 JR      Z,SWT6
3197 CD 72 44              CALL    @MSX
319A CD BA 31              CALL    PRSW
319D CD EE 1F              CALL    #LTNL
31A0 21                    DEFB    21H             ;LD HL,nn
31A1 CB 19         SWT6:   RR      C
31A3 EB            SWT7:   EX      DE,HL
31A4 CD 56 42              CALL    SEAZ
31A7 EB                    EX      DE,HL
31A8 10 E9                 DJNZ    SWT5
31AA C9                    RET
31AB               ;
31AB 43 20 45 46   OPTTBL: DEFB    'C EFLMSK'
31AF 4C 4D 53 4B
31B3               OPTTBE:
31B3               ;
31B3 3A EC 45      AFILE:  LD      A,(FLCNT)
31B6 E6 7F                 AND     07FH
31B8 3D                    DEC     A
31B9 C9                    RET
31BA               ;
31BA C5            PRSW:   PUSH    BC
31BB 06 14                 LD      B,20
31BD CD DF 1F              CALL    #TAB
31C0 C1                    POP     BC
31C1 CD E2 1F              CALL    #MPRNT
31C4 3A 4F 00              DEFB    ':O',0
31C7 3E 6E         PRSW0:  LD      A,'n'
31C9 CB 19                 RR      C
31CB DA F4 1F              JP      C,#PRINT
31CE CD E2 1F              CALL    #MPRNT
31D1 66 66 00              DEFB    'ff',0
31D4 C9                    RET
31D5               ;
31D5               ;       printer switch
31D5               ;
31D5 CD E2 1F      #COM:   CALL    #MPRNT
31D8 50 72 69 6E           DEFB    'Printer O',0
31DC 74 65 72 20
31E0 4F 00
31E2 3A 11 30              LD      A,(LPTSW)
31E5 2F                    CPL
31E6 32 11 30              LD      (LPTSW),A
31E9 4F                    LD      C,A
31EA 18 DB                 JR      PRSW0
31EC               ;
31EC               ;       code area
31EC               ;
31EC CD B2 1F      PCOM:   CALL    #HLHEX
31EF 38 0D                 JR      C,PCOM0
31F1 22 0D 30              LD      (MEMLOW),HL
31F4 CD A6 3F              CALL    SPCUT
31F7 FE 2D                 CP      '-'
31F9 20 03                 JR      NZ,PCOM0
31FB CD A5 3F              CALL    SPCUT0
31FE CD B2 1F      PCOM0:  CALL    #HLHEX
3201 38 03                 JR      C,PCOM1
3203 22 0F 30              LD      (MEMHI),HL
3206 CD 65 44      PCOM1:  CALL    @MPRNT
3209 88                    DEFB    @OBJ
320A 41 72 65 61   @AREMS: DEFB    'Area :',0
320E 20 3A 00
3211 2A 0D 30              LD      HL,(MEMLOW)
3214 CD BE 1F              CALL    #PRTHL
3217 CD 39 32              CALL    PRB
321A 2A 0F 30              LD      HL,(MEMHI)
321D CD BE 1F              CALL    #PRTHL
3220 3E 48                 LD      A,'H'
3222 C3 F4 1F              JP      #PRINT
3225               ;
3225               ;       text area
3225               ;
3225 CD B2 1F      XCOM:   CALL    #HLHEX
3228 38 03                 JR      C,XCOM0
322A 22 08 30              LD      (TXAREA),HL
322D CD 65 44      XCOM0:  CALL    @MPRNT
3230 87 89 00              DEFB    @SOU,@ARE,0
3233 2A 08 30              LD      HL,(TXAREA)
3236 CD BE 1F              CALL    #PRTHL
3239 CD E2 1F      PRB:    CALL    #MPRNT
323C 48 2D 00              DEFB    'H-',0
323F C9                    RET
3240               ;
3240               ;       evaluate expression
3240               ;               and print value
3240               ;
3240 AF            ?COM:   XOR     A
3241 32 F4 45              LD      (EEDFLG),A
3244 3D                    DEC     A
3245 32 EB 45              LD      (PASS),A
3248 3C                    INC     A
3249 D5                    PUSH    DE
324A EB                    EX      DE,HL
324B BE            ?COM0:  CP      (HL)
324C 23                    INC     HL
324D 20 FC                 JR      NZ,?COM0
324F 2B                    DEC     HL
3250 36 0D                 LD      (HL),0DH
3252 D1                    POP     DE
3253 CD E6 3F              CALL    EVAL
3256 69                    LD      L,C
3257 60                    LD      H,B
3258 E5                    PUSH    HL
3259 CD 5D 42              CALL    PRDEC
325C E1                    POP     HL
325D 3E 28                 LD      A,'('
325F CD F4 1F              CALL    #PRINT
3262 CD BE 1F              CALL    #PRTHL
3265 CD E2 1F              CALL    #MPRNT
3268 48 29 00              DEFB    'H)',0
326B C9                    RET
326C               ;
326C               ;       print out label list
326C               ;
326C               OCOM:
326C CD 43 44      CRF:    CALL    LPTON
326F 2A D7 45              LD      HL,(HSHSIZ)
3272 CD 94 1F      CRF0:   CALL    #PEEK
3275 B7                    OR      A
3276 28 45                 JR      Z,@LTNL
3278 CD F4 1F      CRF1:   CALL    #PRINT
327B 23                    INC     HL
327C CD 94 1F              CALL    #PEEK
327F FE 0D                 CP      CR
3281 20 F5                 JR      NZ,CRF1
3283 23                    INC     HL
3284 3E 3A                 LD      A,':'
3286 CD F4 1F              CALL    #PRINT
3289 CD 94 1F              CALL    #PEEK
328C 23                    INC     HL
328D 5F                    LD      E,A
328E CD 94 1F              CALL    #PEEK
3291 23                    INC     HL
3292 57                    LD      D,A
3293 EB                    EX      DE,HL
3294 CD BE 1F              CALL    #PRTHL
3297 CD 18 20              CALL    #CSR
329A 7D                    LD      A,L
329B C6 13                 ADD     A,19
329D 6F                    LD      L,A
329E 26 00                 LD      H,0
32A0 01 14 00              LD      BC,20
32A3 CD A0 42              CALL    DIVC
32A6 CD 86 42              CALL    MUL
32A9 45                    LD      B,L
32AA EB                    EX      DE,HL
32AB CD B5 32              CALL    WCHK
32AE CD C7 1F              CALL    #PAUSE
32B1 BD 32                 DEFW    @LTNL
32B3 18 BD                 JR      CRF0
32B5               ;
32B5 3A 5C 1F      WCHK:   LD      A,(#WIDTH)
32B8 3D                    DEC     A
32B9 B8                    CP      B
32BA D2 DF 1F              JP      NC,#TAB
32BD C3 EE 1F      @LTNL:  JP      #LTNL
32C0               ;
32C0               ;       assemble
32C0               ;
32C0 AF            ACOM:   XOR     A
32C1 32 F1 45              LD      (LSTSW),A
32C4 32 EC 45              LD      (FLCNT),A
32C7 3A 05 30              LD      A,(EDFLG)
32CA 32 F4 45              LD      (EEDFLG),A
32CD 1A                    LD      A,(DE)
32CE D6 2F                 SUB     '/'
32D0 20 05                 JR      NZ,ACOM0
32D2 3D                    DEC     A
32D3 32 F1 45              LD      (LSTSW),A
32D6 13                    INC     DE
32D7 CD A6 3F      ACOM0:  CALL    SPCUT
32DA B7                    OR      A
32DB CA B2 34              JP      Z,ASM
32DE ED 53 ED 45           LD      (FILPT1),DE
32E2 0E 00                 LD      C,0
32E4 0C            ACOM1:  INC     C
32E5 CD 5D 34              CALL    FILE
32E8 F5                    PUSH    AF
32E9 3A 5D 1F              LD      A,(#DSK)
32EC CD 5F 44              CALL    ISTAPE
32EF 20 02                 JR      NZ,ACOM2
32F1 CB F9                 SET     7,C
32F3 F1            ACOM2:  POP     AF
32F4 20 EE                 JR      NZ,ACOM1
32F6 79                    LD      A,C
32F7 32 EC 45              LD      (FLCNT),A
32FA CD 0B 35              CALL    INIASM
32FD CD 20 33      ACOM3:  CALL    APASS
3300 CD 2D 35              CALL    NXPAS
3303 20 0A                 JR      NZ,ACOM4
3305 3A EC 45              LD      A,(FLCNT)
3308 E6 80                 AND     080H
330A C4 EE 33              CALL    NZ,RWAIT
330D 18 EE                 JR      ACOM3
330F 3A 12 30      ACOM4:  LD      A,(OPTFLG)
3312 F6 ED                 OR      0EDH            ;11101101
3314 3C                    INC     A
3315 C0                    RET     NZ              ;S- and/or F-
3316 CD 5D 33              CALL    SAVE?
3319 CD EE 1F              CALL    #LTNL
331C D8                    RET     C
331D C3 87 33              JP      ASAVE0
3320               ;
3320               ;       pass
3320               ;
3320 CD 3B 35      APASS:  CALL    PRPASS
3323 ED 5B ED 45           LD      DE,(FILPT1)
3327 18 0C                 JR      APAS1
3329 3A 12 30      APAS0:  LD      A,(OPTFLG)
332C E6 10                 AND     10H
332E CC 4F 35              CALL    Z,INIAS0        ;F+
3331 ED 5B EF 45           LD      DE,(FILPTR)
3335 1A            APAS1:  LD      A,(DE)
3336 B7                    OR      A
3337 C8                    RET     Z
3338 CD 7A 34              CALL    FLOAD
333B CD 61 35              CALL    TXASM
333E 3A EB 45              LD      A,(PASS)
3341 B7                    OR      A
3342 28 E5                 JR      Z,APAS0         ;pass1
3344 CD C0 34              CALL    ASM2
3347 3A 12 30              LD      A,(OPTFLG)
334A CB 4F                 BIT     1,A
334C 28 DB                 JR      Z,APAS0         ;S-
334E E6 10                 AND     10H
3350 20 D7                 JR      NZ,APAS0        ;F-
3352 CD 5D 33              CALL    SAVE?
3355 CD EE 1F              CALL    #LTNL
3358 D4 8E 33              CALL    NC,ASAVE
335B 18 CC                 JR      APAS0
335D               ;
335D               ;       ask
335D               ;
335D CD 65 44      SAVE?:  CALL    @MPRNT
3360 8A 3F 00              DEFB    @SAV,'?',0
3363 CD 74 33      YNLP:   CALL    KEYIN
3366 FE 0D                 CP      CR
3368 C8                    RET     Z
3369 E6 DF                 AND     0DFH            ;11011111
336B FE 59                 CP      'Y'
336D C8                    RET     Z
336E FE 4E                 CP      'N'
3370 20 F1                 JR      NZ,YNLP
3372 37                    SCF
3373 C9                    RET
3374               ;
3374 CD 21 20      KEYIN:  CALL    #FLGET
3377 FE 1B                 CP      BRK
3379 C0                    RET     NZ
337A               ;
337A               ;       abort
337A               ;
337A CD E2 1F      ABT:    CALL    #MPRNT
337D 0D 42 72 65           DEFB    CR,'Break',0
3381 61 6B 00
3384 C3 2B 30      ABT0:   JP      HOT0
3387               ;
3387               ;       save object in Acom
3387               ;
3387 ED 5B ED 45   ASAVE0: LD      DE,(FILPT1)
338B CD A3 1F              CALL    #FILE
338E DD 22 E4 45   ASAVE:  LD      (OBJST),IX
3392 CD B3 31              CALL    AFILE
3395 28 0D                 JR      Z,ASAVE1
3397 3A 12 30              LD      A,(OPTFLG)
339A E6 10                 AND     10H             ;F+?
339C 20 06                 JR      NZ,ASAVE1
339E 21 FA 1F              LD      HL,#HOT
33A1 22 6E 1F              LD      (#EXADR),HL
```


▶お願いです。これからは浪人生とは呼ばすに「フリーの学生」と呼んでほしいと思います。多分，今年はそういわれそうだから……。　秋山　吉康（18）栃木県

```
33A4 2A 74 1F      ASAVE1: LD      HL,(#IBFAD)
33A7 11 15 30              LD      DE,OBJNAM
33AA 06 0D                 LD      B,13
33AC 23            ASAVE2: INC     HL
33AD 7E                    LD      A,(HL)
33AE FE 0D                 CP      0DH
33B0 20 02                 JR      NZ,ASAVE3
33B2 3E 20                 LD      A,SPC
33B4 12            ASAVE3: LD      (DE),A
33B5 13                    INC     DE
33B6 10 F4                 DJNZ    ASAVE2
33B8 11 13 30              LD      DE,OBJDEV
33BB 1A                    LD      A,(DE)
33BC CD 5C 44              CALL    ISTAPQ
33BF 20 6A                 JR      NZ,SAVE1
33C1 B7                    OR      A
33C2 CD C9 33              CALL    TWAIT
33C5 CD 2B 34              CALL    SAVE1
33C8 37                    SCF
33C9 D9            TWAIT:  EXX
33CA F5                    PUSH    AF
33CB CD E2 1F              CALL    #MPRNT
33CE 53 65 74 20           DEFB    'Set ',0
33D2 00
33D3 F1                    POP     AF
33D4 11 D3 44              LD      DE,OBJMES
33D7 30 03                 JR      NC,TWAIT0
33D9 11 CB 44              LD      DE,SOUMES
33DC CD E5 1F      TWAIT0: CALL    #MSX
33DF CD E2 1F              CALL    #MPRNT
33E2 54 61 70 65           DEFB    'Tape',0
33E6 00
33E7 D9                    EXX
33E8 CD 74 33      TWAIT1: CALL    KEYIN
33EB C3 EE 1F              JP      #LTNL
33EE 11 DB 44      RWAIT:  LD      DE,REWMES
33F1 18 E9                 JR      TWAIT0
33F3               ;
33F3               ;               file error
33F3               ;
33F3 CD 33 20      @ERROR: CALL    #ERROR
33F6 18 8C                 JR      ABT0
33F8               ;
33F8               ;               save
33F8               ;
33F8               SCOM:
33F8 CD 57 34      SAVE:   CALL    GETHEX
33FB D8                    RET     C
33FC 22 70 1F              LD      (#DTADR),HL
33FF 22 E2 45              LD      (SAVADR),HL
3402 4D                    LD      C,L
3403 44                    LD      B,H
3404 CD 57 34              CALL    GETHEX
3407 D8                    RET     C
3408 B7                    OR      A
3409 ED 42                 SBC     HL,BC
340B 23                    INC     HL
340C 22 72 1F              LD      (#SIZE),HL
340F CD 57 34              CALL    GETHEX
3412 D8                    RET     C
3413 22 6E 1F              LD      (#EXADR),HL
3416 CD A6 3F              CALL    SPCUT
3419 FE 3A                 CP      ':'
341B 28 0D                 JR      Z,SAVE0
341D CD B2 1F              CALL    #HLHEX
3420 D8                    RET     C
3421 22 E2 45              LD      (SAVADR),HL
3424 CD A6 3F              CALL    SPCUT
3427 FE 3A                 CP      ':'
3429 C0                    RET     NZ
342A 13            SAVE0:  INC     DE
342B 3E 01         SAVE1:  LD      A,1             ;OBJ
342D CD A3 1F              CALL    #FILE
3430 CD AF 1F      SAVE2:  CALL    #WOPEN
3433 DA F3 33      @@ERR:  JP      C,@ERROR
3436 CD 65 44              CALL    @MPRNT
3439 57 72 69 74           DEFB    'Writ',@ING,0
343D 8B 00
343F CD 38 44              CALL    FPRNT
3442 2A E2 45              LD      HL,(SAVADR)
3445 22 70 1F              LD      (#DTADR),HL
3448 CD AC 1F              CALL    #WRD
344B 38 E6                 JR      C,@@ERR
344D CD E2 1F              CALL    #MPRNT
3450 44 6F 6E 65           DEFB    'Done',CR,0
3454 0D 00
3456 C9                    RET
3457               ;
3457 CD A6 3F      GETHEX: CALL    SPCUT
345A C3 B2 1F              JP      #HLHEX
345D               ;
345D               ;               set filename
345D               ;
345D 3E 04         FILE:   LD      A,4     ;ASC
345F C5                    PUSH    BC
3460 CD A3 1F              CALL    #FILE
3463 C1                    POP     BC
3464 1A                    LD      A,(DE)
3465 FE 3A                 CP      ':'
3467 20 01                 JR      NZ,FILE0
3469 13                    INC     DE
346A CD A6 3F      FILE0:  CALL    SPCUT
346D B7                    OR      A
346E C9                    RET
346F               ;
346F               ;               open file to read
346F               ;
346F CD 09 20      ROPEN:  CALL    #ROPEN
3472 38 BF                 JR      C,@@ERR
3474 C8                    RET     Z
3475 CD 38 44              CALL    FPRNT
3478 18 F5                 JR      ROPEN
347A               ;
347A               ;               load a text
347A               ;
347A CD 5D 34      FLOAD:  CALL    FILE
347D ED 53 EF 45           LD      (FILPTR),DE
3481 3A EB 45              LD      A,(PASS)
3484 B7                    OR      A
3485 28 04                 JR      Z,FLOAD0
3487 CD B3 31              CALL    AFILE
348A C8                    RET     Z
348B CD 6F 34      FLOAD0: CALL    ROPEN
348E CD E2 1F              CALL    #MPRNT
3491 52 65 61 64           DEFB    'Read'
3495 69 6E 67 20   @INGMS: DEFB    'ing ',0
3499 00
349A CD 38 44              CALL    FPRNT
349D 2A 08 30              LD      HL,(TXAREA)
34A0 22 70 1F              LD      (#DTADR),HL
34A3 CD A6 1F              CALL    #RDD
34A6 38 8B         @@@ER:  JR      C,@@ERR
34A8 C9                    RET
34A9               ;
34A9               ;               directory
34A9               ;
34A9 CD A3 1F      DCOM:   CALL    #FILE
34AC CD 06 20              CALL    #DIR
34AF 38 F5                 JR      C,@@@ER
34B1 C9                    RET
34B2               ;
34B2               ;               assemble on memory
34B2               ;
34B2 CD 0B 35      ASM:    CALL    INIASM
34B5 CD 3B 35      ASM0:   CALL    PRPASS
34B8 CD 61 35              CALL    TXASM
34BB CD 2D 35              CALL    NXPAS
34BE 28 F5                 JR      Z,ASM0
34C0 2A D9 45      ASM2:   LD      HL,(LBLMAX)
34C3 ED 5B DD 45           LD      DE,(LBLCNT)
34C7 AF                    XOR     A
34C8 ED 52                 SBC     HL,DE
34CA 3D                    DEC     A
34CB CD 5D 42              CALL    PRDEC
34CE CD 65 44              CALL    @MPRNT
34D1 8C 28 73 29           DEFB    @SLB,'(s)^M::',0
34D5 0D 3A 3A 00
34D9 2A E4 45              LD      HL,(OBJST)
34DC E5                    PUSH    HL
34DD ED 5B CE 45           LD      DE,(OFSADR)
34E1 B7                    OR      A
34E2 ED 52                 SBC     HL,DE
34E4 E5                    PUSH    HL
34E5 22 70 1F              LD      (#DTADR),HL
34E8 CD 84 35              CALL    PRTHLS
34EB 2A D0 45              LD      HL,(LOGADR)
34EE 2B                    DEC     HL
34EF CD 84 35              CALL    PRTHLS
34F2 D1                    POP     DE
34F3 B7                    OR      A
34F4 ED 52                 SBC     HL,DE
34F6 23                    INC     HL
34F7 22 72 1F              LD      (#SIZE),HL
34FA EB                    EX      DE,HL
34FB 22 6E 1F              LD      (#EXADR),HL
34FE CD 84 35              CALL    PRTHLS
3501 E1                    POP     HL
3502 22 E2 45              LD      (SAVADR),HL
3505 CD BE 1F              CALL    #PRTHL
3508 C3 EE 1F              JP      #LTNL
350B               ;
350B FD 21 C0 37   INIASM: LD      IY,PUT3
350F 21 FF FF              LD      HL,-1
3512 22 F2 45              LD      (LSTSW2),HL
3515 CD 8A 35              CALL    INIHSH
3518 AF                    XOR     A
3519 32 EB 45              LD      (PASS),A
351C 21 12 30              LD      HL,OPTFLG
351F CB F6                 SET     6,(HL)          ;pass1
3521 21 D7 45              LD      HL,HSHSIZ
3524 11 DB 45              LD      DE,LBLPTR
3527 01 04 00              LD      BC,4
352A ED B0                 LDIR
352C C9                    RET
352D               ;
352D 21 12 30      NXPAS:  LD      HL,OPTFLG
3530 CB B6                 RES     6,(HL)          ;pass2
3532 3A EB 45              LD      A,(PASS)
3535 3C                    INC     A
3536 32 EB 45              LD      (PASS),A
3539 3D                    DEC     A
353A C9                    RET
353B               ;
353B CD E2 1F      PRPASS: CALL    #MPRNT
353E 50 61 73 73           DEFB    'Pass:',0
3542 3A 00
3544 3A EB 45              LD      A,(PASS)
3547 C6 31                 ADD     A,'1'
3549 CD F4 1F              CALL    #PRINT
354C CD EE 1F              CALL    #LTNL
354F 21 CA 45      INIAS0: LD      HL,LINNO        ;clear  LINNO,ORGADR,
3552 11 CB 45              LD      DE,LINNO+1      ;       OFSADR,LOGADR,
3555 01 09 00              LD      BC,10-1         ;       ORGFLG,OFSFLG
3558 36 00                 LD      (HL),0
355A ED B0                 LDIR
355C DD 21 00 00           LD      IX,0
3560 C9                    RET
3561               ;
3561               ;               assemble a text
3561               ;
3561 ED 5B 08 30   TXASM:  LD      DE,(TXAREA)
3565 1A                    LD      A,(DE)
3566 B7                    OR      A
3567 CA 2B 30              JP      Z,HOT0
356A CD 9C 35      TXASM0: CALL    LINASM
356D D8                    RET     C
356E 3A EB 45              LD      A,(PASS)
3571 B7                    OR      A
3572 28 F6                 JR      Z,TXASM0
3574 2A F1 45              LD      HL,(LSTSW)
3577 7C                    LD      A,H
3578 A5                    AND     L
3579 C4 82 43              CALL    NZ,PRLST
357C 3A F3 45              LD      A,(LSTSW3)
357F 32 F2 45              LD      (LSTSW2),A
3582 18 E6                 JR      TXASM0
3584               ;
3584 CD BE 1F      PRTHLS: CALL    #PRTHL
3587 C3 F1 1F              JP      #PRNTS
358A               ;
358A               ;               init hash table
358A               ;
358A 21 00 00      INIHSH: LD      HL,0
358D ED 4B D7 45           LD      BC,(HSHSIZ)
3591 03                    INC     BC
3592 AF                    XOR     A
3593 CD 9A 1F      INIHS0: CALL    #POKE
3596 ED A1                 CPI
3598 EA 93 35              JP      PE,INIHS0
359B C9                    RET
359C               ;
359C               ;               assemble a line
359C               ;
359C ED 53 C8 45   LINASM: LD      (LINPTR),DE
35A0 AF                    XOR     A
35A1 32 D4 45              LD      (LBLFLG),A
35A4 1A                    LD      A,(DE)
35A5 B7                    OR      A
35A6 37                    SCF
35A7 C8                    RET     Z
35A8 2A D0 45              LD      HL,(LOGADR)
35AB 22 E6 45              LD      (PADR),HL
35AE DD 22 E8 45           LD      (LOBPTR),IX
35B2 2A CA 45              LD      HL,(LINNO)
35B5 23                    INC     HL
35B6 22 CA 45              LD      (LINNO),HL
35B9 FE 09                 CP      TAB
35BB 28 20                 JR      Z,LNASM0
35BD FE 3B                 CP      ';'
35BF 28 61                 JR      Z,REMCUT
35C1 FE 0D                 CP      CR
35C3 28 68                 JR      Z,NXLIN
35C5 FE 20                 CP      SPC
35C7 28 14                 JR      Z,LNASM0
35C9 FE 2E                 CP      '.'
35CB CA 30 36              JP      Z,@COM
35CE CD EC 36              CALL    ISLB1ST
35D1 D2 13 43              JP      NC,ERR3
35D4 CD 35 36              CALL    DEFLBL
35D7 1A                    LD      A,(DE)
35D8 FE 3A                 CP      ':'
35DA 28 01                 JR      Z,LNASM0
35DC 1B                    DEC     DE
35DD CD A5 3F      LNASM0: CALL    SPCUT0
35E0 FE 3B                 CP      ';'
35E2 28 3E                 JR      Z,REMCUT
35E4 FE 0D                 CP      CR
35E6 28 45                 JR      Z,NXLIN
35E8 FE 2E                 CP      '.'
35EA 28 44                 JR      Z,@COM
35EC D6 41                 SUB     'A'
35EE 38 16                 JR      C,ERR40
35F0 FE 1A                 CP      'Z'-'A'+1
35F2 3F                    CCF
35F3 38 11                 JR      C,ERR40
35F5 87                    ADD     A,A
35F6 4F                    LD      C,A
35F7 06 00                 LD      B,0
35F9 21 E6 37              LD      HL,C0TBL
35FC 09                    ADD     HL,BC
35FD 7E                    LD      A,(HL)
35FE 23                    INC     HL
35FF 66                    LD      H,(HL)
3600 6F                    LD      L,A
3601 EB            LNASM1: EX      DE,HL
3602 CD 14 37              CALL    SEAOP
3605 EB                    EX      DE,HL
3606 DA 16 43      ERR40:  JP      C,ERR4
3609 CD A6 3F              CALL    SPCUT
360C ED 4B 12 30           LD      BC,(OPTFLG)     ;C=opt flags
3610 C5                    PUSH    BC
3611 F1                    POP     AF
```

▶またもや，しばしば，しょっちゅう，いつもどおりといおうか，マンマと広告に乗せられて，ふと気づくと机の上にはPC-E500とカラッポの財布。またやったか，と思う間もなく素早く立ち直り，怪しげなジャノメ基板を挟んでX1 データレコーダがつながったりしている。あー，早く清く正しい一般人になりたい。　木本　忠雄　(20)　大阪府

```
3612 08                     EX      AF,AF'          ;F'=opt flags
3613 CD 3B 37               CALL    GENCOD
3616 3A 12 30               LD      A,(OPTFLG)
3619 CB 77                  BIT     6,A
361B 20 05                  JR      NZ,REMCUT       ;pass1
361D E6 20                  AND     20H
361F C4 02 43               CALL    NZ,EOLCHK       ;E+
3622 EB            REMCUT:  EX      DE,HL
3623 01 00 00               LD      BC,0
3626 3E 0D                  LD      A,CR
3628 ED B1                  CPIR
362A EB                     EX      DE,HL
362B B7                     OR      A
362C C9                     RET                     ;NC
362D 13            NXLIN:   INC     DE
362E B7                     OR      A               ;NC
362F C9                     RET
3630               ;
3630 21 18 39      @COM:    LD      HL,@TBL
3633 18 CC                  JR      LNASM1
3635               ;
3635               ;        define a label
3635               ;
3635 21 D4 45      DEFLBL:  LD      HL,LBLFLG
3638 36 01                  LD      (HL),1
363A D5                     PUSH    DE
363B CD F8 36               CALL    HASH
363E E5            DFLBL0:  PUSH    HL              ;hash
363F D9                     EXX
3640 E1                     POP     HL              ;hash
3641 D1                     POP     DE              ;strptr
3642 3A EB 45               LD      A,(PASS)
3645 B7                     OR      A
3646 28 17                  JR      Z,DFLBL1
3648 CD B2 36               CALL    ISDEF
364B DA 0D 43               JP      C,ERR1
364E ED 43 D5 45            LD      (LSTLBL),BC
3652 2A DD 45               LD      HL,(LBLCNT)
3655 7C                     LD      A,H
3656 B5                     OR      L
3657 28 56                  JR      Z,@ER6
3659 2B                     DEC     HL
365A 22 DD 45               LD      (LBLCNT),HL
365D D9                     EXX
365E C9                     RET
365F CD B2 36      DFLBL1:  CALL    ISDEF
3662 D2 10 43               JP      NC,ERR2
3665 ED 4B DB 45            LD      BC,(LBLPTR)
3669 79                     LD      A,C
366A CD 9A 1F               CALL    #POKE
366D 23                     INC     HL
366E 78                     LD      A,B
366F CD 9A 1F               CALL    #POKE
3672 69                     LD      L,C
3673 60                     LD      H,B
3674 08                     EX      AF,AF'
3675 CA 1F 43               JP      Z,ERR7          ;label len = 0
3678 47                     LD      B,A
3679 1A            DFLBL2:  LD      A,(DE)
367A CD 9A 1F               CALL    #POKE
367D 13                     INC     DE
367E 23                     INC     HL
367F 10 F8                  DJNZ    DFLBL2
3681 3E 0D                  LD      A,CR
3683 CD 9A 1F               CALL    #POKE
3686 23                     INC     HL
3687 22 D5 45               LD      (LSTLBL),HL
368A ED 5B D0 45            LD      DE,(LOGADR)
368E 7B                     LD      A,E
368F CD 9A 1F               CALL    #POKE
3692 23                     INC     HL
3693 7A                     LD      A,D
3694 CD 9A 1F               CALL    #POKE
3697 23                     INC     HL
3698 AF                     XOR     A
3699 CD 9A 1F               CALL    #POKE
369C 22 DB 45               LD      (LBLPTR),HL
369F 3A 12 30               LD      A,(OPTFLG)
36A2 E6 08                  AND     08H
36A4 D9                     EXX
36A5 C8                     RET     Z               ;L-
36A6 D9                     EXX
36A7 ED 5B E0 45            LD      DE,(WKMAX)
36AB ED 52                  SBC     HL,DE
36AD D9                     EXX
36AE D8                     RET     C
36AF C3 1C 43      @ER6:    JP      ERR6
36B2               ;
36B2               ;        check whether defined the label
36B2               ;        in      DE=strptr
36B2               ;                HL=hash
36B2               ;                A'=label len
36B2               ;
36B2 3A DF 45      ISDEF:   LD      A,(HSHMSK)
36B5 A4                     AND     H
36B6 67                     LD      H,A
36B7 CD 94 1F               CALL    #PEEK
36BA 4F                     LD      C,A
36BB 23                     INC     HL
36BC CD 94 1F               CALL    #PEEK
36BF 2B                     DEC     HL
36C0 47                     LD      B,A
36C1 B1                     OR      C
36C2 37                     SCF
36C3 C8                     RET     Z               ;not defined
36C4 D5                     PUSH    DE
36C5 E5                     PUSH    HL
36C6 69                     LD      L,C
36C7 60                     LD      H,B
36C8 08                     EX      AF,AF'
36C9 47                     LD      B,A             ;B=lebel len
36CA 08                     EX      AF,AF'
36CB CD 94 1F      ISDEF0:  CALL    #PEEK
36CE EB                     EX      DE,HL
36CF BE                     CP      (HL)
36D0 EB                     EX      DE,HL
36D1 20 12                  JR      NZ,ISDEF1
36D3 23                     INC     HL
36D4 13                     INC     DE
36D5 10 F4                  DJNZ    ISDEF0
36D7 CD 94 1F               CALL    #PEEK
36DA FE 0D                  CP      CR
36DC 20 07                  JR      NZ,ISDEF1
36DE 23                     INC     HL
36DF 4D                     LD      C,L
36E0 44                     LD      B,H             ;val adr
36E1 E1                     POP     HL              ;hash
36E2 D1                     POP     DE              ;strptr
36E3 B7                     OR      A               ;NC
36E4 C9                     RET                     ;defined
36E5 E1            ISDEF1:  POP     HL
36E6 D1                     POP     DE
36E7 23                     INC     HL
36E8 23                     INC     HL
36E9 C3 B2 36               JP      ISDEF
36EC               ;
36EC CD 9E 3F      ISLB1ST:CALL     ISDEC
36EF D0                     RET     NC
36F0 FE F4                  CP      '$'-'0'
36F2 C8                     RET     Z
36F3 FE F5                  CP      '%'-'0'
36F5 C8                     RET     Z
36F6 37                     SCF
36F7 C9                     RET
36F8               ;
36F8               ;        calculate hash of label
36F8               ;
36F8 21 00 00      HASH:    LD      HL,0
36FB 08                     EX      AF,AF'
36FC AF                     XOR     A               ;A=label len
36FD 08            HASH0:   EX      AF,AF'
36FE CD B9 3F               CALL    ISLSPR0
3701 C8                     RET     Z
3702 4D                     LD      C,L
3703 44                     LD      B,H
3704 29                     ADD     HL,HL
3705 29                     ADD     HL,HL
3706 09                     ADD     HL,BC
3707 4F                     LD      C,A
3708 06 00                  LD      B,0
370A 09                     ADD     HL,BC
370B 29                     ADD     HL,HL
370C 13                     INC     DE
370D 08                     EX      AF,AF'
370E 3C                     INC     A
370F 20 EC                  JR      NZ,HASH0
3711 C3 13 43               JP      ERR3
3714               ;
3714               ;        search opecode
3714               ;
3714 23            SEAOP:   INC     HL
3715 4D                     LD      C,L
3716 44                     LD      B,H
3717 1A            SEAOP0:  LD      A,(DE)
3718 B7                     OR      A
3719 37                     SCF
371A C8                     RET     Z               ;CY=1
371B 1A            SEAOP1:  LD      A,(DE)
371C 13                     INC     DE
371D B7                     OR      A
371E FA 28 37               JP      M,SEAOP2
3721 BE                     CP      (HL)
3722 20 0C                  JR      NZ,SEAOP3
3724 23                     INC     HL
3725 C3 1B 37               JP      SEAOP1
3728 08            SEAOP2:  EX      AF,AF'
3729 7E                     LD      A,(HL)
372A CD D4 3F               CALL    ISBLNK
372D 20 07                  JR      NZ,SEAOP4
372F C9                     RET                     ;NC,AF'=no
3730 1A            SEAOP3:  LD      A,(DE)
3731 13                     INC     DE
3732 B7                     OR      A
3733 F2 30 37               JP      P,SEAOP3
3736 69            SEAOP4:  LD      L,C
3737 60                     LD      H,B
3738 C3 17 37               JP      SEAOP0
373B               ;
373B               ;        generate object code
373B               ;
373B E6 7F         GENCOD:  AND     7FH
373D FE 10                  CP      10H
373F 38 11                  JR      C,GEN1
3741 D6 30                  SUB     30H
3743 38 49                  JR      C,GEN2
3745 01 5B 39               LD      BC,C3TBL
3748 6F            TBLJMP:  LD      L,A
3749 26 00                  LD      H,0
374B 29                     ADD     HL,HL
374C 09                     ADD     HL,BC
374D 4E                     LD      C,(HL)
374E 23                     INC     HL
374F 66                     LD      H,(HL)
3750 69                     LD      L,C
3751 E9                     JP      (HL)
3752               ;
3752               ;        1byte code
3752               ;
3752 4F            GEN1:    LD      C,A
3753 08                     EX      AF,AF'
3754 28 21                  JR      Z,PUT11         ;pass1
3756 06 00                  LD      B,0
3758 21 24 39               LD      HL,C1TBL
375B 09                     ADD     HL,BC
375C 7E                     LD      A,(HL)
375D 18 0D                  JR      PUT10
375F               ;
375F 3E CB         PUTCB:   LD      A,0CBH
3761 18 05                  JR      PUT1
3763 3E ED         PUTED:   LD      A,0EDH
3765 18 01                  JR      PUT1
3767 79            PUTC:    LD      A,C
3768 08            PUT1:    EX      AF,AF'
3769 28 0C                  JR      Z,PUT11         ;pass1
376B 2E                     DEFB    2EH             ;LD L,n
376C 08            PUT10:   EX      AF,AF'
376D EC C5 37               CALL    PE,MEMCHK       ;M+
3770 08                     EX      AF,AF'
3771 DD 77 00               LD      (IX),A
3774 DD 23                  INC     IX
3776 2E                     DEFB    2EH             ;LD L,n
3777 08            PUT11:   EX      AF,AF'
3778 2A D0 45               LD      HL,(LOGADR)
377B 23                     INC     HL
377C 22 D0 45               LD      (LOGADR),HL
377F C9                     RET
3780               ;
3780 FE 0F         PUT[X]:  CP      15
3782 18 02                  JR      PUTIX0
3784 FE 0D         PUTIX:   CP      13
3786 3E DD         PUTIX0:  LD      A,0DDH
3788 28 DE                  JR      Z,PUT1
378A 3E FD                  LD      A,0FDH
378C 18 DA                  JR      PUT1
378E               ;
378E               ;        2bytes code
378E               ;
378E 08            GEN2:    EX      AF,AF'
378F 28 25                  JR      Z,PUT21         ;pass1
3791 08                     EX      AF,AF'
3792 C6 20                  ADD     A,20H
3794 87                     ADD     A,A
3795 4F                     LD      C,A
3796 06 00                  LD      B,0
3798 21 31 39               LD      HL,C2TBL
379B 09                     ADD     HL,BC
379C 4E                     LD      C,(HL)
379D 23                     INC     HL
379E 46                     LD      B,(HL)
379F 18 05                  JR      PUT20
37A1               ;
37A1 4F            PUTAB:   LD      C,A
37A2 08            PUT2:    EX      AF,AF'
37A3 28 11                  JR      Z,PUT21         ;pass1
37A5 2E                     DEFB    2EH             ;LD L,n
37A6 08            PUT20:   EX      AF,AF'
37A7 EC C5 37               CALL    PE,MEMCHK       ;M+
37AA 08                     EX      AF,AF'
37AB DD 71 00               LD      (IX),C
37AE DD 23                  INC     IX
37B0 DD 70 00               LD      (IX),B
37B3 DD 23                  INC     IX
37B5 2E                     DEFB    2EH             ;LD L,n
37B6 08            PUT21:   EX      AF,AF'
37B7 2A D0 45               LD      HL,(LOGADR)
37BA 23                     INC     HL
37BB 23                     INC     HL
37BC 22 D0 45               LD      (LOGADR),HL
37BF C9                     RET
37C0               ;
37C0 CD 68 37      PUT3:    CALL    PUT1
37C3 18 DD                  JR      PUT2
37C5               ;
37C5               ;        check memory
37C5               ;
37C5 08            MEMCHK:  EX      AF,AF'
37C6 F5                     PUSH    AF
37C7 D5                     PUSH    DE
37C8 DD E5                  PUSH    IX
37CA E1                     POP     HL
37CB ED 5B 0D 30            LD      DE,(MEMLOW)
37CF E5                     PUSH    HL
37D0 B7                     OR      A
37D1 ED 52                  SBC     HL,DE
37D3 E1                     POP     HL
37D4 D1                     POP     DE
37D5 DA 37 43               JP      C,ERR15         ;mem protected
37D8 D5                     PUSH    DE
37D9 ED 5B 0F 30            LD      DE,(MEMHI)
37DD ED 52                  SBC     HL,DE
37DF D1                     POP     DE
37E0 D2 37 43               JP      NC,ERR15        ;mem protected
```


▶12月号では，沙羅曼蛇の攻略法を考えたり，ラスト・ハルマゲドンを紹介してくれたりしてとても凄いと思った。スゴロクの内容が面白いので，できたら毎月やってほしい。

野尻　哲　(13)　愛媛県

```
37E3 F1                     POP     AF
37E4 08                     EX      AF,AF'
37E5 C9                     RET
37E6               ;
37E6               ;        opecode table
37E6               ;
37E6 16 38 20 38   C0TBL:   DEFW    ATBL,BTBL
37EA 24 38 3F 38            DEFW    CTBL,DTBL
37EE 64 38 23 39            DEFW    ETBL,FTBL
37F2 23 39 6F 38            DEFW    GTBL,HTBL
37F6 74 38 8A 38            DEFW    ITBL,JTBL
37FA 23 39 8F 38            DEFW    KTBL,LTBL
37FE 23 39 A0 38            DEFW    MTBL,NTBL
3802 A7 38 C6 38            DEFW    OTBL,PTBL
3806 23 39 CE 38            DEFW    QTBL,RTBL
380A FE 38 23 39            DEFW    STBL,TTBL
380E 23 39 23 39            DEFW    UTBL,VTBL
3812 23 39 14 39            DEFW    WTBL,XTBL
3816               ;
0080 P             n        EQU     80H
3816               ;
3816 44 44 B0      ATBL:    DEFB    'DD',    30H+n
3819 4E 44 B1               DEFB    'ND',    31H+n
381C 44 43 B2               DEFB    'DC',    32H+n
381F 00                     DEFB    0
3820 49 54 B3      BTBL:    DEFB    'IT',    33H+n
3823 00                     DEFB    0
3824 50 B4         CTBL:    DEFB    'P',     34H+n
3826 41 4C 4C B5            DEFB    'ALL',   35H+n
382A 43 46 80               DEFB    'CF',    00H+n
382D 50 4C 81               DEFB    'PL',    01H+n
3830 50 49 90               DEFB    'PI',    10H+n
3833 50 44 91               DEFB    'PD',    11H+n
3836 50 49 52 92            DEFB    'PIR',   12H+n
383A 50 44 52 93            DEFB    'PDR',   13H+n
383E 00                     DEFB    0
383F 45 43 B6      DTBL:    DEFB    'EC',    36H+n
3842 4A 4E 5A B7            DEFB    'JNZ',   37H+n
3846 45 46 42 B8            DEFB    'EFB',   38H+n
384A 45 46 57 B9            DEFB    'EFW',   39H+n
384E 45 46 53 BA            DEFB    'EFS',   3AH+n
3852 42 B8                  DEFB    'B',     38H+n
3854 57 B9                  DEFB    'W',     39H+n
3856 53 BA                  DEFB    'S',     3AH+n
3858 41 41 83               DEFB    'AA',    03H+n
385B 49 82                  DEFB    'I',     02H+n
385D 4D BB                  DEFB    'M',     3BH+n
385F 45 46 4D BB            DEFB    'EFM',   3BH+n
3863 00                     DEFB    0
3864 58 BC         ETBL:    DEFB    'X',     3CH+n
3866 58 58 84               DEFB    'XX',    04H+n
3869 51 55 BD               DEFB    'QU',    3DH+n
386C 49 85                  DEFB    'I',     05H+n
386E 00                     DEFB    0
386F 41 4C 54 86   HTBL:    DEFB    'ALT',   06H+n
3873 00                     DEFB    0
3874 4E 43 BE      ITBL:    DEFB    'NC',    3EH+n
3877 4E BF                  DEFB    'N',     3FH+n
3879 4E 49 94               DEFB    'NI',    14H+n
387C 4E 44 95               DEFB    'ND',    15H+n
387F 4E 49 52 96            DEFB    'NIR',   16H+n
3883 4E 44 52 97            DEFB    'NDR',   17H+n
3887 4D C0                  DEFB    'M',     40H+n
3889 00                     DEFB    0
388A 52 C1         JTBL:    DEFB    'R',     41H+n
388C 50 C2                  DEFB    'P',     42H+n
388E 00                     DEFB    0
388F 44 C3         LTBL:    DEFB    'D',     43H+n
3891 44 49 52 98            DEFB    'DIR',   18H+n
3895 44 49 99               DEFB    'DI',    19H+n
3898 44 44 52 9A            DEFB    'DDR',   1AH+n
389C 44 44 9B               DEFB    'DD',    1BH+n
389F 00                     DEFB    0
38A0 45 47 9C      NTBL:    DEFB    'EG',    1CH+n
38A3 4F 50 87               DEFB    'OP',    07H+n
38A6 00                     DEFB    0
38A7 52 C4         OTBL:    DEFB    'R',     44H+n
38A9 55 54 C5               DEFB    'UT',    45H+n
38AC 55 54 49 9D            DEFB    'UTI',   1DH+n
38B0 54 49 52 9E            DEFB    'TIR',   1EH+n
38B4 55 54 44 9F            DEFB    'UTD',   1FH+n
38B8 54 44 52 A0            DEFB    'TDR',   20H+n
38BC 52 47 C6               DEFB    'RG',    46H+n
38BF 46 46 53 45            DEFB    'FFSET',47H+n
38C3 54 C7
38C5 00                     DEFB    0
38C6 4F 50 C8      PTBL:    DEFB    'OP',    48H+n
38C9 55 53 48 C9            DEFB    'USH',   49H+n
38CD 00                     DEFB    0
38CE 45 54 CA      RTBL:    DEFB    'ET',    4AH+n
38D1 4C 41 88               DEFB    'LA',    08H+n
38D4 52 41 89               DEFB    'RA',    09H+n
38D7 4C 43 41 8A            DEFB    'LCA',   0AH+n
38DB 52 43 41 8B            DEFB    'RCA',   0BH+n
38DF 4C CB                  DEFB    'L',     4BH+n
38E1 52 CC                  DEFB    'R',     4CH+n
38E3 4C 43 CD               DEFB    'LC',    4DH+n
38E6 52 43 CE               DEFB    'RC',    4EH+n
38E9 4C 44 A1               DEFB    'LD',    21H+n
38EC 52 44 A2               DEFB    'RD',    22H+n
38EF 45 53 CF               DEFB    'ES',    4FH+n
38F2 53 54 D0               DEFB    'ST',    50H+n
38F5 45 54 49 A3            DEFB    'ETI',   23H+n
38F9 45 54 4E A4            DEFB    'ETN',   24H+n
38FD 00                     DEFB    0
38FE 42 43 D1      STBL:    DEFB    'BC',    51H+n
3901 55 42 D2               DEFB    'UB',    52H+n
3904 43 46 8C               DEFB    'CF',    0CH+n
3907 45 54 D3               DEFB    'ET',    53H+n
390A 4C 41 D4               DEFB    'LA',    54H+n
390D 52 41 D5               DEFB    'RA',    55H+n
3910 52 4C D6               DEFB    'RL',    56H+n
3913 00                     DEFB    0
3914 4F 52 D7      XTBL:    DEFB    'OR',    57H+n
3917 00                     DEFB    0
3918 4C 49 53 54   @TBL:    DEFB    'LIST',  58H+n
391C D8
391D 4E 4C 49 53            DEFB    'NLIST',59H+n
3921 54 D9
3923               FTBL:
3923               GTBL:
3923               KTBL:
3923               MTBL:
3923               QTBL:
3923               TTBL:
3923               UTBL:
3923               VTBL:
3923               WTBL:
3923 00                     DEFB    0
3924               ;
3924 3F            C1TBL:   CCF                              ;00
3925 2F                     CPL                              ;01
3926 F3                     DI                               ;02
3927 27                     DAA                              ;03
3928 D9                     EXX                              ;04
3929 FB                     EI                               ;05
392A 76                     HALT                             ;06
392B 00                     NOP                              ;07
392C 17                     RLA                              ;08
392D 1F                     RRA                              ;09
392E 07                     RLCA                             ;0A
392F 0F                     RRCA                             ;0B
3930 37                     SCF                              ;0C
3931               ;
3931 ED A1         C2TBL:   CPI                              ;10
3933 ED A9                  CPD                              ;11
3935 ED B1                  CPIR                             ;12
3937 ED B9                  CPDR                             ;13
3939 ED A2                  INI                              ;14
393B ED AA                  IND                              ;15
393D ED B2                  INIR                             ;16
393F ED BA                  INDR                             ;17
3941 ED B0                  LDIR                             ;18
3943 ED A0                  LDI                              ;19
3945 ED B8                  LDDR                             ;1A
3947 ED A8                  LDD                              ;1B
3949 ED 44                  NEG                              ;1C
394B ED A3                  OUTI                             ;1D
394D ED B3                  OTIR                             ;1E
394F ED AB                  OUTD                             ;1F
3951 ED BB                  OTDR                             ;20
3953 ED 6F                  RLD                              ;21
3955 ED 67                  RRD                              ;22
3957 ED 4D                  RETI                             ;23
3959 ED 45                  RETN                             ;24
395B               ;
395B BC 3B 7B 3B   C3TBL:   DEFW    #ADD,    #AND     ;30 31
395F 25 3C A6 3D            DEFW    #ADC,    #BIT     ;32 33
3963 8A 3B F2 3C            DEFW    #CP,     #CALL    ;34 35
3967 43 3C 8F 3C            DEFW    #DEC,    #DJNZ    ;36 37
396B BA 3E D5 3E            DEFW    #DEFB,   #DEFW    ;38 39
396F 62 3F E1 3E            DEFW    #DEFS,   #DM      ;3A 3B
3973 6B 3B 94 3E            DEFW    #EX,     #EQU     ;3C 3D
3977 69 3C 23 3D            DEFW    #INC,    #IN      ;3E 3F
397B 1D 3E 93 3C            DEFW    #IM,     #JR      ;40 41
397F D4 3C 15 3A            DEFW    #JP,     #LD      ;42 43
3983 87 3B 6E 3D            DEFW    #OR,     #OUT     ;44 45
3987 4F 3E 7F 3E            DEFW    #ORG,    #OFFSET  ;46 47
398B EF 3B EC 3B            DEFW    #POP,    #PUSH    ;48 49
398F 11 3D CE 3D            DEFW    #RET,    #RL      ;4A 4B
3993 CB 3D C8 3D            DEFW    #RR,     #RLC     ;4C 4D
3997 C5 3D A0 3D            DEFW    #RRC,    #RES     ;4E 4F
399B FC 3D 16 3C            DEFW    #RST,    #SBC     ;50 51
399F 81 3B A3 3D            DEFW    #SUB,    #SET     ;52 53
39A3 BF 3D BC 3D            DEFW    #SLA,    #SRA     ;54 55
39A7 C2 3D 7E 3B            DEFW    #SRL,    #XOR     ;56 57
39AB 35 3E 48 3E            DEFW    #LIST,   #NLIST   ;58 59
39AF               ;
39AF 85 3A 85 3A   LDTBL:   DEFW    LD16,LD16
39B3 A5 3A 68 3A            DEFW    LDHL,LDSP
39B7 19 43 10 3B            DEFW    ERR5,LD8
39BB 10 3B 10 3B            DEFW    LD8,LD8
39BF 10 3B 10 3B            DEFW    LD8,LD8
39C3 10 3B 10 3B            DEFW    LD8,LD8
39C7 BF 3A A2 3A            DEFW    LDA,LDIX
39CB A2 3A 4D 3B            DEFW    LDIX,LD[X]
39CF 4D 3B                  DEFW    LD[X]
39D1               ;
39D1 53 3C 53 3C   DECTBL:  DEFW    DEC16,DEC16
39D5 53 3C 53 3C            DEFW    DEC16,DEC16
39D9 19 43 4C 3C            DEFW    ERR5,DEC8
39DD 4C 3C 4C 3C            DEFW    DEC8,DEC8
39E1 4C 3C 4C 3C            DEFW    DEC8,DEC8
39E5 4C 3C 4C 3C            DEFW    DEC8,DEC8
39E9 4C 3C 58 3C            DEFW    DEC8,DECIX
39ED 58 3C 60 3C            DEFW    DECIX,DEC[X]
39F1 60 3C                  DEFW    DEC[X]
39F3               ;
39F3 79 3C 79 3C   INCTBL:  DEFW    INC16,INC16
39F7 79 3C 79 3C            DEFW    INC16,INC16
39FB 19 43 72 3C            DEFW    ERR5,INC8
39FF 72 3C 72 3C            DEFW    INC8,INC8
3A03 72 3C 72 3C            DEFW    INC8,INC8
3A07 72 3C 72 3C            DEFW    INC8,INC8
3A0B 72 3C 7E 3C            DEFW    INC8,INCIX
3A0F 7E 3C 86 3C            DEFW    INCIX,INC[X]
3A13 86 3C                  DEFW    INC[X]
3A15               ;
3A15 CD 3C 41      #LD:     CALL    GETREG
3A18 38 0B                  JR      C,LD[]
3A1A FE 0F                  CP      15
3A1C DC F3 42               CALL    C,CMACHK
3A1F 01 AF 39               LD      BC,LDTBL
3A22 C3 48 37               JP      TBLJMP
3A25               ;
3A25 21 E4 41      LD[]:    LD      HL,[]TBL
3A28 CD B0 41               CALL    CMPGEN
3A2B D0                     RET     NC
3A2C CD EC 42               CALL    [EVAL]
3A2F C5                     PUSH    BC
3A30 CD F3 42               CALL    CMACHK
3A33 CD 35 41               CALL    REGCHK
3A36 FE 0C                  CP      12              ;A?
3A38 CA 51 3A               JP      Z,LD[]A
3A3B FE 04                  CP      4
3A3D 38 17                  JR      C,LD[]16
3A3F FE 0D                  CP      13
3A41 DA 19 43      ERR5C:   JP      C,ERR5
3A44 FE 0F                  CP      15
3A46 3F                     CCF
3A47 38 F8                  JR      C,ERR5C
3A49 CD 84 37      LD[]IX:  CALL    PUTIX
3A4C C1            LD[]HL:  POP     BC
3A4D 3E 22                  LD      A,22H           ;00 100 010
3A4F FD E9                  JP      (IY)            ;PUT3
3A51               ;
3A51 3E 32         LD[]A:   LD      A,32H           ;00 110 010
3A53 C1                     POP     BC
3A54 FD E9                  JP      (IY)            ;PUT3
3A56               ;
3A56 FE 02         LD[]16:  CP      2               ;HL?
3A58 28 F2                  JR      Z,LD[]HL
3A5A 87                     ADD     A,A
3A5B 87                     ADD     A,A
3A5C 87                     ADD     A,A
3A5D 87                     ADD     A,A
3A5E F6 43                  OR      43H             ;01 XX0 011
3A60 F5                     PUSH    AF
3A61 CD 63 37               CALL    PUTED
3A64 F1                     POP     AF
3A65 C1                     POP     BC
3A66 FD E9                  JP      (IY)            ;PUT3
3A68               ;
3A68 CD 3C 41      LDSP:    CALL    GETREG
3A6B 38 16                  JR      C,LDSP0
3A6D 0E F9                  LD      C,0F9H
3A6F FE 02                  CP      2               ;HL?
3A71 CA 67 37               JP      Z,PUTC
3A74 FE 0D                  CP      13              ;IX?
3A76 01 DD F9               LD      BC,0F9DDH
3A79 CA A2 37               JP      Z,PUT2
3A7C FE 0E                  CP      14              ;IY?
3A7E 0E FD                  LD      C,0FDH
3A80 CA A2 37               JP      Z,PUT2
3A83 3E 03         LDSP0:   LD      A,3
3A85                        ;
3A85 87            LD16:    ADD     A,A
3A86 87                     ADD     A,A
3A87 87                     ADD     A,A
3A88 87                     ADD     A,A
3A89 F5                     PUSH    AF
3A8A CD FC 42               CALL    IS[
3A8D 28 08                  JR      Z,LD[16]
3A8F CD E6 3F               CALL    EVAL
3A92 F1                     POP     AF
3A93 F6 01                  OR      01H             ;00 XX0 001
3A95 FD E9                  JP      (IY)            ;PUT3
3A97               ;
3A97 CD 63 37      LD[16]:  CALL    PUTED
3A9A CD E2 42               CALL    EVAL]
3A9D F1                     POP     AF
3A9E F6 4B                  OR      4BH             ;01 XX1 011
3AA0 FD E9                  JP      (IY)            ;PUT3
3AA2               ;
3AA2 CD 84 37      LDIX:    CALL    PUTIX
3AA5 CD FC 42      LDHL:    CALL    IS[
3AA8 28 07                  JR      Z,LDHL[]
3AAA CD E6 3F               CALL    EVAL
3AAD 3E 21                  LD      A,21H           ;LD HL,nn
3AAF FD E9                  JP      (IY)            ;PUT3
3AB1 CD E2 42      LDHL[]:  CALL    EVAL]
3AB4 3E 2A                  LD      A,2AH           ;LD HL,(nn)
3AB6 FD E9                  JP      (IY)            ;PUT3
3AB8               ;
3AB8 D6 05         LDA8:    SUB     5
3ABA F6 78                  OR      78H             ;01 111 000
3ABC C3 68 37               JP      PUT1
3ABF               ;
3ABF CD 3C 41      LDA:     CALL    GETREG
3AC2 38 12                  JR      C,LDA0
3AC4 FE 05                  CP      5               ;rp?
3AC6 38 55                  JR      C,ER5C0
```


▶12月15日の「とんねるずのみなさんのおかげです」のサンダーバードに名機MZ-2000が出ていたのを見て，つくづく世の中は諸行無常だと思った。　有山　剛史　(17)　埼玉県

```
3AC8 FE 0D              CP      13          ;reg?
3ACA 38 EC              JR      C,LDA8
3ACC FE 0F              CP      15          ;index reg?
3ACE 38 4D              JR      C,ER5C0
3AD0 06 38     LDA[X]:  LD      B,38H
3AD2 C5                 PUSH    BC
3AD3 C3 3F 3B           JP      LD8[X]0
3AD6           ;
3AD6 CD FC 42  LDA0:    CALL    IS[
3AD9 28 27              JR      Z,LDA[]
3ADB FE 49              CP      'I'
3ADD 28 18              JR      Z,LDAI
3ADF FE 52              CP      'R'
3AE1 20 0B              JR      NZ,LDAN
3AE3 13        LDAR:    INC     DE
3AE4 CD D3 3F           CALL    ISBLK0
3AE7 01                 DEFB    01H         ;LD BC,nn
3AE8 ED 5F              LD      A,R
3AEA CA A2 37           JP      Z,PUT2
3AED 1B        LDAR0:   DEC     DE
3AEE CD E6 3F  LDAN:    CALL    EVAL
3AF1 41                 LD      B,C
3AF2 0E 3E              LD      C,3EH       ;LD A,n
3AF4 C3 A2 37  LDAN0:   JP      PUT2
3AF7 13        LDAI:    INC     DE
3AF8 CD D3 3F           CALL    ISBLK0
3AFB 20 F0              JR      NZ,LDAR0
3AFD 01                 DEFB    01H         ;LD BC,nn
3AFE ED 57              LD      A,I
3B00 18 F2              JR      LDAN0
3B02           ;
3B02 21 03 42  LDA[]:   LD      HL,A[]TBL
3B05 CD B0 41           CALL    CMPGEN
3B08 D0                 RET     NC
3B09 CD E2 42           CALL    EVAL]
3B0C 3E 3A              LD      A,3AH       ;LD A,(nn)
3B0E FD E9              JP      (IY)        ;PUT3
3B10           ;
3B10 D6 05     LD8:     SUB     5
3B12 87                 ADD     A,A
3B13 87                 ADD     A,A
3B14 87                 ADD     A,A
3B15 F5                 PUSH    AF
3B16 CD 3C 41           CALL    GETREG
3B19 38 16              JR      C,LD8N
3B1B FE 05              CP      5           ;rp?
3B1D 38 0F     ER5C0:   JR      C,ERR50
3B1F FE 0D              CP      13          ;index?
3B21 30 18              JR      NC,LD8[X]
3B23 D6 05              SUB     5
3B25 C1                 POP     BC
3B26 B0                 OR      B
3B27 F6 40              OR      40H         ;01 XXX XXX
3B29 FE 76              CP      76H         ;01 110 110
3B2B C2 68 37           JP      NZ,PUT1
3B2E C3 19 43  ERR50:   JP      ERR5
3B31 CD E6 3F  LD8N:    CALL    EVAL
3B34 F1                 POP     AF
3B35 F6 06              OR      06H         ;00 XXX 110
3B37 41                 LD      B,C
3B38 C3 A1 37           JP      PUTAB
3B3B           ;
3B3B FE 0F     LD8[X]:  CP      15          ;index reg?
3B3D 38 EF              JR      C,ERR50
3B3F CD D1 42  LD8[X]0:CALL     PUT[X]D
3B42 41                 LD      B,C
3B43 F1                 POP     AF
3B44 FE 30              CP      30H         ;XXX LD (HL),(IX+d)
3B46 28 E6              JR      Z,ERR50
3B48 F6 46              OR      46H         ;01 000 110
3B4A C3 A1 37           JP      PUTAB
3B4D           ;
3B4D CD D1 42  LD[X]:   CALL    PUT[X]D
3B50 CD F3 42           CALL    CMACHK
3B53 79                 LD      A,C
3B54 F5                 PUSH    AF          ;A=d
3B55 CD 43 41           CALL    GETRG8
3B58 C1                 POP     BC          ;B=d
3B59 38 05              JR      C,LD[X]N
3B5B F6 70              OR      70H         ;01 110 XXX
3B5D C3 A1 37  LD[X]0:  JP      PUTAB
3B60 0E 36     LD[X]N:  LD      C,36H
3B62 CD A2 37           CALL    PUT2
3B65 CD E6 3F           CALL    EVAL
3B68 C3 67 37           JP      PUTC
3B6B           ;
3B6B 21 10 42  #EX:     LD      HL,EXTBL
3B6E CD B0 41           CALL    CMPGEN
3B71 D0                 RET     NC
3B72 C3 19 43  ERR51:   JP      ERR5
3B75           ;
3B75 0E 88     ADCA:    LD      C,88H       ;10 001 000
3B77 21                 DEFB    21H
3B78 0E 98     SBCA:    LD      C,98H       ;10 011 000
3B7A 21                 DEFB    21H
3B7B 0E A0     #AND:    LD      C,0A0H      ;10 100 000
3B7D 21                 DEFB    21H
3B7E 0E A8     #XOR:    LD      C,0A8H      ;10 101 000
3B80 21                 DEFB    21H
3B81 0E 90     #SUB:    LD      C,90H       ;10 010 000
3B83 21                 DEFB    21H
3B84 0E 80     ADDA:    LD      C,80H       ;10 000 000
3B86 21                 DEFB    21H
3B87 0E B0     #OR:     LD      C,0B0H      ;10 110 000
3B89 21                 DEFB    21H
3B8A 0E B8     #CP:     LD      C,0B8H      ;10 111 000
3B8C                    ;
3B8C CD 3C 41           CALL    GETREG
3B8F 38 10              JR      C,CPN
3B91 FE 0F              CP      15          ;(IX+d)?
3B93 30 18              JR      NC,CP[X]
3B95 D6 05              SUB     5
3B97 38 D9              JR      C,ERR51     ;rp
3B99 FE 08              CP      8
3B9B 30 D5              JR      NC,ERR51    ;index reg
3B9D B1                 OR      C
3B9E C3 68 37           JP      PUT1
3BA1 79        CPN:     LD      A,C
3BA2 F5                 PUSH    AF
3BA3 CD E6 3F           CALL    EVAL
3BA6 41                 LD      B,C
3BA7 F1                 POP     AF
3BA8 F6 C6              OR      0C6H        ;11 000 110
3BAA C3 A1 37           JP      PUTAB
3BAD           ;
3BAD CD 80 37  CP[X]:   CALL    PUT[X]
3BB0 3E 86              LD      A,086H      ;10 000 110
3BB2 B1                 OR      C
3BB3 CD 68 37           CALL    PUT1
3BB6 CD D4 42           CALL    GETIXD
3BB9 C3 67 37           JP      PUTC
3BBC           ;
3BBC CD 35 41  #ADD:    CALL    REGCHK
3BBF CD F3 42           CALL    CMACHK
3BC2 FE 0C              CP      12          ;A?
3BC4 28 BE              JR      Z,ADDA
3BC6 FE 02              CP      2           ;HL?
3BC8 28 17              JR      Z,ADDHL
3BCA FE 0D              CP      13          ;index reg?
3BCC 38 A4              JR      C,ERR51
3BCE FE 0F              CP      15          ;(IX+d)?
3BD0 30 A0              JR      NC,ERR51
3BD2 F5        ADDIX:   PUSH    AF
3BD3 CD 84 37           CALL    PUTIX
3BD6 CD 35 41           CALL    REGCHK
3BD9 C1                 POP     BC
3BDA B8                 CP      B
3BDB 20 07              JR      NZ,ADDHL0
3BDD 3E 02              LD      A,02H       ;HL
3BDF 18 07              JR      ADDHL1
3BE1 CD 35 41  ADDHL:   CALL    REGCHK
3BE4 FE 04     ADDHL0:  CP      4           ;rp?
3BE6 30 8A              JR      NC,ERR51    ;
3BE8 0E 09     ADDHL1:  LD      C,09H       ;00 XX1 001
3BEA 18 22              JR      PUSH1
3BEC           ;
3BEC 0E C5     #PUSH:   LD      C,0C5H      ;11 XX0 101
3BEE 21                 DEFB    21H
3BEF 0E C1     #POP:    LD      C,0C1H      ;11 XX0 001
3BF1 CD 35 41           CALL    REGCHK
3BF4 FE 03              CP      3           ;SP?
3BF6 28 48              JR      Z,ERR52     ;XXX POP SP
3BF8 38 14              JR      C,PUSH1
3BFA FE 04              CP      4           ;AF?
3BFC 28 0F              JR      Z,PUSH0
3BFE FE 0D              CP      13          ;reg?
3C00 38 3E              JR      C,ERR52
3C02 FE 0F              CP      15          ;index reg?
3C04 30 3A              JR      NC,ERR52
3C06 CD 84 37           CALL    PUTIX
3C09 3E 02              LD      A,02H       ;HL
3C0B 18 01              JR      PUSH1
3C0D 3D        PUSH0:   DEC     A           ;SP
3C0E 87        PUSH1:   ADD     A,A
3C0F 87        PUSH2:   ADD     A,A
3C10 87                 ADD     A,A
3C11 87                 ADD     A,A
3C12 B1                 OR      C
3C13 C3 68 37           JP      PUT1
3C16           ;
3C16 CD 35 41  #SBC:    CALL    REGCHK
3C19 CD F3 42           CALL    CMACHK
3C1C FE 0C              CP      12          ;A?
3C1E CA 78 3B           JP      Z,SBCA
3C21 0E 42              LD      C,42H       ;01 XX0 010
3C23 18 0D              JR      ADC0
3C25           ;
3C25 CD 35 41  #ADC:    CALL    REGCHK
3C28 CD F3 42           CALL    CMACHK
3C2B FE 0C              CP      12          ;A?
3C2D CA 75 3B           JP      Z,ADCA
3C30 0E 4A              LD      C,4AH       ;01 XX1 010
3C32 FE 02     ADC0:    CP      2           ;HL?
3C34 20 0A              JR      NZ,ERR52    ;
3C36 CD 63 37           CALL    PUTED
3C39 CD 35 41           CALL    REGCHK
3C3C FE 04              CP      4           ;rp?
3C3E 38 CE              JR      C,PUSH1
3C40 C3 19 43  ERR52:   JP      ERR5
3C43           ;
3C43 CD 35 41  #DEC:    CALL    REGCHK
3C46 01 D1 39           LD      BC,DECTBL
3C49 C3 48 37           JP      TBLJMP
3C4C           ;
3C4C D6 05     DEC8:    SUB     5
3C4E 0E 05              LD      C,05H       ;00 XXX 101
3C50 C3 0F 3C           JP      PUSH2
3C53           ;
3C53 0E 0B     DEC16:   LD      C,0BH       ;00 XX1 011
3C55 C3 0E 3C           JP      PUSH1
3C58           ;
3C58 CD 84 37  DECIX:   CALL    PUTIX
3C5B 3E 2B              LD      A,2BH       ;00 101 011
3C5D C3 68 37           JP      PUT1
3C60           ;
3C60 CD D1 42  DEC[X]:  CALL    PUT[X]D
3C63 41                 LD      B,C
3C64 0E 35              LD      C,35H       ;00 110 101
3C66 C3 A2 37           JP      PUT2
3C69           ;
3C69 CD 35 41  #INC:    CALL    REGCHK
3C6C 01 F3 39           LD      BC,INCTBL
3C6F C3 48 37           JP      TBLJMP
3C72           ;
3C72 D6 05     INC8:    SUB     5
3C74 0E 04              LD      C,04H       ;00 XXX 100
3C76 C3 0F 3C           JP      PUSH2
3C79           ;
3C79 0E 03     INC16:   LD      C,03H       ;00 XX0 011
3C7B C3 0E 3C           JP      PUSH1
3C7E           ;
3C7E CD 84 37  INCIX:   CALL    PUTIX
3C81 3E 23              LD      A,23H       ;00 100 011
3C83 C3 68 37           JP      PUT1
3C86           ;
3C86 CD D1 42  INC[X]:  CALL    PUT[X]D
3C89 41                 LD      B,C
3C8A 0E 34              LD      C,34H       ;00 110 100
3C8C C3 A2 37           JP      PUT2
3C8F           ;
3C8F 0E 10     #DJNZ:   LD      C,10H       ;DJNZ d
3C91 18 14              JR      JR0
3C93           ;
3C93 CD 2E 41  #JR:     CALL    GETCND
3C96 0E 18              LD      C,18H       ;JR d
3C98 38 0D              JR      C,JR0
3C9A FE 04              CP      4
3C9C 30 A2              JR      NC,ERR52    ;XXX PO,PE,P,M
3C9E CD F3 42           CALL    CMACHK
3CA1 87                 ADD     A,A
3CA2 87                 ADD     A,A
3CA3 87                 ADD     A,A
3CA4 F6 20              OR      20H         ;00 1XX 000
3CA6 4F                 LD      C,A
3CA7 CD 67 37  JR0:     CALL    PUTC
3CAA CD E6 3F  JR1:     CALL    EVAL
3CAD 08                 EX      AF,AF'
3CAE CA 77 37           JP      Z,PUT11     ;pass1
3CB1 08        JR2:     EX      AF,AF'
3CB2 69                 LD      L,C
3CB3 60                 LD      H,B
3CB4 ED 4B D0 45        LD      BC,(LOGADR)
3CB8 03                 INC     BC
3CB9 B7                 OR      A
3CBA ED 42              SBC     HL,BC
3CBC 38 0B              JR      C,JR3
3CBE 25                 DEC     H
3CBF 24                 INC     H
3CC0 20 0F              JR      NZ,JR4
3CC2 7D                 LD      A,L
3CC3 B7                 OR      A
3CC4 F2 68 37           JP      P,PUT1
3CC7 18 08              JR      JR4
3CC9 24        JR3:     INC     H
3CCA 20 05              JR      NZ,JR4
3CCC 7D                 LD      A,L
3CCD B7                 OR      A
3CCE FA 68 37           JP      M,PUT1
3CD1 C3 22 43  JR4:     JP      ERR8        ;too far
3CD4           ;
3CD4 CD 2E 41  #JP:     CALL    GETCND
3CD7 0E C2              LD      C,0C2H      ;11 XXX 010
3CD9 30 1E              JR      NC,CALL1
3CDB 21 40 42           LD      HL,JPTBL
3CDE CD B0 41           CALL    CMPGEN
3CE1 D0                 RET     NC
3CE2 3E C3              LD      A,0C3H      ;JP nn
3CE4 CD 68 37           CALL    PUT1
3CE7 08                 EX      AF,AF'
3CE8 CA B6 37           JP      Z,PUT21     ;pass1
3CEB 08                 EX      AF,AF'
3CEC CD E6 3F           CALL    EVAL
3CEF C3 A6 37           JP      PUT20
3CF2           ;
3CF2 CD 2E 41  #CALL:   CALL    GETCND
3CF5 38 0A              JR      C,CALL2
3CF7 0E C4     CALL0:   LD      C,0C4H      ;11 XXX 100
3CF9 CD 0F 3C  CALL1:   CALL    PUSH2
3CFC CD F3 42           CALL    CMACHK
3CFF 18 05              JR      CALL3
3D01 3E CD     CALL2:   LD      A,0CDH      ;CALL nn
3D03 CD 68 37           CALL    PUT1
3D06 08        CALL3:   EX      AF,AF'
3D07 CA B6 37           JP      Z,PUT21     ;pass1
3D0A 08                 EX      AF,AF'
3D0B CD E6 3F           CALL    EVAL
3D0E C3 A6 37           JP      PUT20
3D11           ;
3D11 CD D3 3F  #RET:    CALL    ISBLK0
3D14 3E C9              LD      A,0C9H      ;RET
3D16 CA 68 37           JP      Z,PUT1
3D19 CD 2E 41           CALL    GETCND
```


▶真夜中にOh!Xを読んでいたら，突然，「宇宙戦艦ヤマト」の曲が聞きたくなった(理由はわからない)。んでもって，アニキのカセットボックスを探していたら，出るわ出るわ懐かしのカセットテープ。さだまさし，YMO，ゴダイゴ，極めつけピンクレディー!! アニキは，いまでは語るも恥ずかしいといっている。 伊藤 史彦 (16) 東京都

```
3D1C 38 29              JR      C,ERR53
3D1E 0E C0              LD      C,0C0H          ;11 XXX 000
3D20 C3 0F 3C           JP      PUSH2
3D23            ;
3D23 CD 3C 41   #IN:    CALL    GETREG
3D26 38 1F              JR      C,ERR53
3D28 CD F3 42           CALL    CMACHK
3D2B D6 05              SUB     5
3D2D 38 18              JR      C,ERR53         ;rp
3D2F FE 06              CP      6               ;(HL)?
3D31 28 14              JR      Z,ERR53
3D33 FE 07              CP      7               ;A?
3D35 28 13              JR      Z,INA
3D37 30 0E              JR      NC,ERR53
3D39 47         INR:    LD      B,A
3D3A CD 63 37           CALL    PUTED
3D3D 78                 LD      A,B
3D3E 0E 40              LD      C,40H           ;01 XXX 000
3D40 CD 0F 3C           CALL    PUSH2
3D43 CD 5A 3D           CALL    IS[C]
3D46 C8                 RET     Z
3D47 C3 19 43   ERR53:  JP      ERR5
3D4A            ;
3D4A 21 67 3D   INA:    LD      HL,[C]
3D4D CD B0 41           CALL    CMPGEN
3D50 D0                 RET     NC
3D51 CD EC 42           CALL    [EVAL]
3D54 41                 LD      B,C
3D55 0E DB              LD      C,0DBH          ;IN A,(n)
3D57 C3 A2 37           JP      PUT2
3D5A            ;
3D5A 21 67 3D   IS[C]:  LD      HL,[C]
3D5D 06 03              LD      B,3
3D5F 1A         IS[C]0: LD      A,(DE)
3D60 BE                 CP      (HL)
3D61 C0                 RET     NZ
3D62 13                 INC     DE
3D63 23                 INC     HL
3D64 10 F9              DJNZ    IS[C]0
3D66 C9                 RET
3D67            ;
3D67 28 43 29 00 [C]:   DEFB    '(C)',0
3D6B ED 78              IN      A,(C)
3D6D 00                 DEFB    0
3D6E            ;
3D6E D5         #OUT:   PUSH    DE
3D6F CD 5A 3D           CALL    IS[C]
3D72 28 16              JR      Z,OUT[C]
3D74 D1                 POP     DE
3D75 CD EC 42           CALL    [EVAL]
3D78 CD F3 42           CALL    CMACHK
3D7B CD 43 41           CALL    GETRG8
3D7E 38 1D              JR      C,ERR54
3D80 FE 07              CP      7               ;A?
3D82 20 19              JR      NZ,ERR54
3D84 41                 LD      B,C
3D85 0E D3              LD      C,0D3H          ;OUT (n),A
3D87 C3 A2 37           JP      PUT2
3D8A            ;
3D8A C1         OUT[C]: POP     BC
3D8B CD 63 37           CALL    PUTED
3D8E CD F3 42           CALL    CMACHK
3D91 CD 43 41           CALL    GETRG8
3D94 38 07              JR      C,ERR54
3D96 FE 06              CP      6               ;B-L?
3D98 0E 41              LD      C,41H           ;01 XXX  001
3D9A C2 0F 3C           JP      NZ,PUSH2        ;A
3D9D C3 19 43   ERR54:  JP      ERR5            ;XXX OUT (C),(HL)
3DA0            ;
3DA0 0E 80      #RES:   LD      C,80H           ;10 XXX XXX
3DA2 21                 DEFB    21H
3DA3 0E C0      #SET:   LD      C,0C0H          ;11 XXX XXX
3DA5 21                 DEFB    21H
3DA6 0E 40      #BIT:   LD      C,40H           ;01 XXX XXX
3DA8 CD 9D 3F           CALL    ISDEC0
3DAB 38 F0              JR      C,ERR54
3DAD FE 08              CP      8               ;0-7?
3DAF 30 EC              JR      NC,ERR54
3DB1 87                 ADD     A,A
3DB2 87                 ADD     A,A
3DB3 87                 ADD     A,A
3DB4 B1                 OR      C
3DB5 4F                 LD      C,A
3DB6 13                 INC     DE
3DB7 CD F3 42           CALL    CMACHK
3DBA 18 14              JR      RL0
3DBC            ;
3DBC 0E 28      #SRA:   LD      C,28H           ;00 101 XXX
3DBE 21                 DEFB    21H
3DBF 0E 20      #SLA:   LD      C,20H           ;00 100 XXX
3DC1 21                 DEFB    21H
3DC2 0E 38      #SRL:   LD      C,38H           ;00 111 XXX
3DC4 21                 DEFB    21H
3DC5 0E 08      #RRC:   LD      C,08H           ;00 001 XXX
3DC7 21                 DEFB    21H
3DC8 0E 00      #RLC:   LD      C,00H           ;00 000 XXX
3DCA 21                 DEFB    21H
3DCB 0E 18      #RR:    LD      C,18H           ;00 011 XXX
3DCD 21                 DEFB    21H
3DCE 0E 10      #RL:    LD      C,10H           ;00 010 XXX
3DD0 CD 35 41   RL0:    CALL    REGCHK
3DD3 FE 0F              CP      15              ;(IX+d)?
3DD5 30 11              JR      NC,RL[X]
3DD7 D6 05              SUB     5
3DD9 38 C2              JR      C,ERR54         ;rp
3DDB FE 08              CP      8
3DDD 30 BE              JR      NC,ERR54        ;index reg
3DDF 47                 LD      B,A
3DE0 CD 5F 37   RLR:    CALL    PUTCB
3DE3 78                 LD      A,B
3DE4 B1                 OR      C
3DE5 C3 68 37           JP      PUT1
3DE8            ;
3DE8 C5         RL[X]:  PUSH    BC
3DE9 CD 80 37           CALL    PUT[X]
3DEC CD 5F 37           CALL    PUTCB
3DEF CD D4 42           CALL    GETIXD
3DF2 CD 67 37           CALL    PUTC
3DF5 C1                 POP     BC
3DF6 79                 LD      A,C
3DF7 F6 06              OR      06H             ;00 XXX 110
3DF9 C3 68 37           JP      PUT1
3DFC            ;
3DFC CD E6 3F   #RST:   CALL    EVAL
3DFF 05                 DEC     B
3E00 04                 INC     B
3E01 20 0F              JR      NZ,ERR55
3E03 79                 LD      A,C
3E04 FE 08              CP      08H
3E06 38 0D              JR      C,RST0
3E08 4F                 LD      C,A
3E09 E6 F8              AND     0F8H            ;XXXXX000?
3E0B B9                 CP      C               ;
3E0C 20 04              JR      NZ,ERR55
3E0E FE 40              CP      40H             ;00-38H?
3E10 38 06              JR      C,RST1
3E12 C3 19 43   ERR55:  JP      ERR5
3E15 87         RST0:   ADD     A,A
3E16 87                 ADD     A,A
3E17 87                 ADD     A,A
3E18 F6 C7      RST1:   OR      0C7H            ;11 000 111
3E1A C3 68 37           JP      PUT1
3E1D            ;
3E1D 1A         #IM:    LD      A,(DE)
3E1E 13                 INC     DE
3E1F D6 30              SUB     '0'
3E21 28 0A              JR      Z,IM0
3E23 38 ED              JR      C,ERR55
3E25 FE 03              CP      03H             ;1-2?
3E27 30 E9              JR      NC,ERR55
3E29 3C                 INC     A
3E2A 87                 ADD     A,A
3E2B 87                 ADD     A,A
3E2C 87                 ADD     A,A
3E2D F6 46      IM0:    OR      046H            ;01 000 110
3E2F 47                 LD      B,A
3E30 0E ED              LD      C,0EDH
3E32 C3 A2 37           JP      PUT2
3E35            ;
3E35 3A F1 45   #LIST:  LD      A,(LSTSW)
3E38 32 F3 45           LD      (LSTSW3),A
3E3B CD D3 3F           CALL    ISBLK0
3E3E C8                 RET     Z
3E3F CD E6 3F           CALL    EVAL
3E42 0B                 DEC     BC
3E43 ED 43 CA 45        LD      (LINNO),BC
3E47 C9                 RET
3E48            ;
3E48 21 00 00   #NLIST: LD      HL,0
3E4B 22 F2 45           LD      (LSTSW2),HL     ;LSTSW2,3 off
3E4E C9                 RET
3E4F            ;
3E4F CD E6 3F   #ORG:   CALL    EVAL
3E52 2A D0 45           LD      HL,(LOGADR)
3E55 ED 43 CC 45        LD      (ORGADR),BC
3E59 ED 43 D0 45        LD      (LOGADR),BC
3E5D ED 43 E6 45        LD      (PADR),BC
3E61 37                 SCF
3E62 ED 42              SBC     HL,BC
3E64 D2 34 43           JP      NC,ERR14
3E67 2A CE 45           LD      HL,(OFSADR)
3E6A 09                 ADD     HL,BC
3E6B 22 E8 45           LD      (LOBPTR),HL
3E6E E5                 PUSH    HL
3E6F DD E1              POP     IX
3E71 3A D2 45           LD      A,(ORGFLG)
3E74 B7                 OR      A
3E75 C0                 RET     NZ
3E76 22 E4 45           LD      (OBJST),HL
3E79 3E FF              LD      A,-1
3E7B 32 D2 45           LD      (ORGFLG),A
3E7E C9                 RET
3E7F            ;
3E7F 2A D2 45   #OFFSET:LD      HL,(ORGFLG)
3E82 7C                 LD      A,H
3E83 B5                 OR      L
3E84 C2 31 43           JP      NZ,ERR13
3E87 3E FF              LD      A,-1
3E89 32 D3 45           LD      (OFSFLG),A
3E8C CD E6 3F           CALL    EVAL
3E8F ED 43 CE 45        LD      (OFSADR),BC
3E93 C9                 RET
3E94            ;
3E94 21 D4 45   #EQU:   LD      HL,LBLFLG
3E97 35                 DEC     (HL)
3E98 34                 INC     (HL)
3E99 CA B0 3E           JP      Z,EQU0
3E9C 34                 INC     (HL)
3E9D CD E6 3F           CALL    EVAL
3EA0 ED 43 E6 45        LD      (PADR),BC
3EA4 2A D5 45           LD      HL,(LSTLBL)
3EA7 79                 LD      A,C
3EA8 CD 9A 1F           CALL    #POKE
3EAB 23                 INC     HL
3EAC 78                 LD      A,B
3EAD C3 9A 1F           JP      #POKE
3EB0 ED 5B C8 45 EQU0:  LD      DE,(LINPTR)
3EB4 C3 1F 43           JP      ERR7
3EB7            ;
3EB7 CD A5 3F   DEFB00: CALL    SPCUT0
3EBA 1A         #DEFB:  LD      A,(DE)
3EBB CD E0 3F           CALL    ISQ
3EBE 20 05              JR      NZ,DEFB0
3EC0 CD E8 3E           CALL    DM0
3EC3 18 07              JR      DEFB1
3EC5 CD E6 3F   DEFB0:  CALL    EVAL
3EC8 79                 LD      A,C
3EC9 CD 68 37           CALL    PUT1
3ECC CD B0 3F   DEFB1:  CALL    ISSPRT
3ECF 28 E6              JR      Z,DEFB00
3ED1 C9                 RET
3ED2            ;
3ED2 CD A5 3F   DEFW00: CALL    SPCUT0
3ED5 CD E6 3F   #DEFW:  CALL    EVAL
3ED8 CD A2 37           CALL    PUT2
3EDB CD B0 3F           CALL    ISSPRT
3EDE 28 F2              JR      Z,DEFW00
3EE0 C9                 RET
3EE1            ;
3EE1 1A         #DM:    LD      A,(DE)
3EE2 CD E0 3F           CALL    ISQ
3EE5 C2 2E 43           JP      NZ,ERR12
3EE8 08         DM0:    EX      AF,AF'
3EE9 38 29              JR      C,DMK0          ;K+
3EEB 08                 EX      AF,AF'
3EEC            ;                       put string if K-
3EEC 13         DM1:    INC     DE
3EED 1A                 LD      A,(DE)
3EEE FE 0D              CP      CR
3EF0 CA 2E 43           JP      Z,ERR12
3EF3 FE 5E              CP      '^'
3EF5 28 18              JR      Z,DM4
3EF7 CD E0 3F           CALL    ISQ
3EFA 20 0E              JR      NZ,DM3
3EFC 13                 INC     DE
3EFD 4F                 LD      C,A
3EFE 1A                 LD      A,(DE)
3EFF CD E0 3F           CALL    ISQ
3F02 28 04              JR      Z,DM2
3F04 CD BA 3F           CALL    ISLSPR
3F07 C8                 RET     Z
3F08 1B         DM2:    DEC     DE
3F09 79                 LD      A,C
3F0A CD 68 37   DM3:    CALL    PUT1
3F0D 18 DD              JR      DM1
3F0F CD 51 3F   DM4:    CALL    ESC
3F12 18 F6              JR      DM3
3F14            ;                       put string if K+
3F14 08         DMK0:   EX      AF,AF'
3F15 13         DMK1:   INC     DE
3F16 1A                 LD      A,(DE)
3F17 FE 0D              CP      CR
3F19 CA 2E 43           JP      Z,ERR12
3F1C FE 5E              CP      '^'
3F1E 28 2C              JR      Z,DMK5
3F20 CD 8F 3F           CALL    ISKNJ
3F23 38 0F              JR      C,DMK2
3F25 CD 68 37           CALL    PUT1
3F28 13                 INC     DE
3F29 1A                 LD      A,(DE)
3F2A FE 0D              CP      CR
3F2C CA 2E 43           JP      Z,ERR12
3F2F CD 68 37           CALL    PUT1
3F32 18 E1              JR      DMK1
3F34 CD E0 3F   DMK2:   CALL    ISQ
3F37 20 0E              JR      NZ,DMK4
3F39 13                 INC     DE
3F3A 4F                 LD      C,A
3F3B 1A                 LD      A,(DE)
3F3C CD E0 3F           CALL    ISQ
3F3F 28 04              JR      Z,DMK3
3F41 CD BA 3F           CALL    ISLSPR
3F44 C8                 RET     Z
3F45 1B         DMK3:   DEC     DE
3F46 79                 LD      A,C
3F47 CD 68 37   DMK4:   CALL    PUT1
3F4A 18 C9              JR      DMK1
3F4C CD 51 3F   DMK5:   CALL    ESC
3F4F 18 F6              JR      DMK4
3F51            ;
3F51 13         ESC:    INC     DE
3F52 1A                 LD      A,(DE)
3F53 FE 0D              CP      CR
3F55 CA 2E 43           JP      Z,ERR12
3F58 D6 40              SUB     '@'
3F5A 38 03              JR      C,ESC0
3F5C FE 1C              CP      '['-'@'+1
3F5E D8                 RET     C
3F5F C6 40      ESC0:   ADD     A,'@'
3F61 C9                 RET
3F62            ;
3F62 CD E6 3F   #DEFS:  CALL    EVAL
```

THE SENTINEL

▶サンダーフォースIIの８方向スクロール攻略法。"VERY HARDで３分間遊ぶ"。そうすると，EASYランクの敵が止まって見えます。　西川　信一　(22)　三重県

```
3F65 78                       LD      A,B
3F66 B1                       OR      C
3F67 CA 25 43                 JP      Z,ERR9
3F6A DD 09                    ADD     IX,BC
3F6C 2A D0 45                 LD      HL,(LOGADR)
3F6F 09                       ADD     HL,BC
3F70 22 D0 45                 LD      (LOGADR),HL
3F73 ED 4B E8 45              LD      BC,(LOBPTR)
3F77 DD 22 E8 45              LD      (LOBPTR),IX
3F7B 08                       EX      AF,AF'
3F7C C8                       RET     Z                ;pass1
3F7D F0                       RET     P                ;C-
3F7E DD E5                    PUSH    IX
3F80 E1                       POP     HL
3F81 AF                       XOR     A
3F82 ED 42                    SBC     HL,BC
3F84 E5                       PUSH    HL
3F85 C5                       PUSH    BC
3F86 E1                       POP     HL
3F87 C1                       POP     BC
3F88 77               DEFS0:  LD      (HL),A
3F89 ED A1                    CPI
3F8B EA 88 3F                 JP      PE,DEFS0
3F8E C9                       RET
3F8F                  ;
3F8F                  ;       check whether kanji 1st char
3F8F                  ;
3F8F FE 81            ISKNJ:  CP      81H
3F91 D8                       RET     C
3F92 FE A0                    CP      0A0H
3F94 3F                       CCF
3F95 D0                       RET     NC
3F96 FE E0                    CP      0E0H
3F98 D8                       RET     C
3F99 FE F0                    CP      0F0H
3F9B 3F                       CCF
3F9C C9                       RET
3F9D                  ;
3F9D                  ;       check whether decimal char
3F9D                  ;
3F9D 1A               ISDEC0: LD      A,(DE)
3F9E D6 30            ISDEC:  SUB     '0'
3FA0 D8                       RET     C
3FA1 FE 0A                    CP      9+1
3FA3 3F                       CCF
3FA4 C9                       RET
3FA5                  ;
3FA5                  ;       skip white space
3FA5                  ;
3FA5 13               SPCUT0: INC     DE
3FA6 1A               SPCUT:  LD      A,(DE)
3FA7 FE 09                    CP      TAB
3FA9 28 FA                    JR      Z,SPCUT0
3FAB FE 20                    CP      SPC
3FAD 28 F6                    JR      Z,SPCUT0
3FAF C9                       RET
3FB0                  ;
3FB0                  ;       check whether separator
3FB0                  ;
3FB0 CD A6 3F         ISSPRT: CALL    SPCUT
3FB3 FE 2C                    CP      ','
3FB5 C8                       RET     Z
3FB6 FE 3A                    CP      ':'
3FB8 C9                       RET
3FB9                  ;
3FB9 1A               ISLSPR0:LD      A,(DE)
3FBA D9               ISLSPR: EXX
3FBB 21 C5 3F                 LD      HL,LSPTBL
3FBE 01 0E 00                 LD      BC,SPREND-LSPTBL
3FC1 ED B1                    CPIR
3FC3 D9                       EXX
3FC4 C9                       RET
3FC5                  ;
3FC5 09 20            LSPTBL: DEFB    TAB,SPC
3FC7 0D 3B 3A 29      ESPTBL: DEFB    CR,';:),+-*/%^"^''
3FCB 2C 2B 2D 2A
3FCF 2F 25 22 27
3FD3                  SPREND:
3FD3                  ;
3FD3                  ;       check whether blank
3FD3                  ;
3FD3 1A               ISBLK0: LD      A,(DE).
3FD4 FE 09            ISBLNK: CP      TAB
3FD6 C8                       RET     Z
3FD7 FE 0D                    CP      CR
3FD9 C8                       RET     Z
3FDA FE 3B                    CP      ';'
3FDC C8                       RET     Z
3FDD FE 20                    CP      SPC
3FDF C9                       RET
3FE0                  ;
3FE0                  ;       check whether quotation mark
3FE0                  ;
3FE0 FE 27            ISQ:    CP      '''
3FE2 C8                       RET     Z
3FE3 FE 22                    CP      '"'
3FE5 C9                       RET
3FE6                  ;
3FE6                  ;       evaluate expression
3FE6                  ;
3FE6 CD 0C 40         EVAL:   CALL    EVAL10
3FE9 CD A6 3F         EVAL0:  CALL    SPCUT
3FEC FE 2B                    CP      '+'
3FEE 20 08                    JR      NZ,EVAL1
3FF0 E5                       PUSH    HL
3FF1 CD 0B 40                 CALL    EVAL9
3FF4 C1                       POP     BC
3FF5 09                       ADD     HL,BC
3FF6 18 F1                    JR      EVAL0
3FF8 FE 2D            EVAL1:  CP      '-'
3FFA 20 0C                    JR      NZ,EVAL2
3FFC E5                       PUSH    HL
3FFD CD 0B 40                 CALL    EVAL9
4000 4D                       LD      C,L
4001 44                       LD      B,H
4002 E1                       POP     HL
4003 B7                       OR      A
4004 ED 42                    SBC     HL,BC
4006 18 E1                    JR      EVAL0
4008 4D               EVAL2:  LD      C,L
4009 44                       LD      B,H
400A C9                       RET
400B                  ;
400B 13               EVAL9:  INC     DE
400C CD 66 40         EVAL10: CALL    EVAL20
400F CD A6 3F         EVAL11: CALL    SPCUT
4012 FE 2A                    CP      '*'
4014 20 0A                    JR      NZ,EVAL12
4016 E5                       PUSH    HL
4017 CD 65 40                 CALL    EVAL19
401A C1                       POP     BC
401B CD 86 42                 CALL    MUL
401E 18 EF                    JR      EVAL11
4020 FE 2F            EVAL12: CP      '/'
4022 20 0C                    JR      NZ,EVAL13
4024 E5                       PUSH    HL
4025 CD 65 40                 CALL    EVAL19
4028 4D                       LD      C,L
4029 44                       LD      B,H
402A E1                       POP     HL
402B CD 99 42                 CALL    DIV
402E 18 DF                    JR      EVAL11
4030 FE 25            EVAL13: CP      '%'
4032 C0                       RET     NZ
4033 E5                       PUSH    HL
4034 CD 65 40                 CALL    EVAL19
4037 4D                       LD      C,L
4038 44                       LD      B,H
4039 E1                       POP     HL
403A D5                       PUSH    DE
403B CD B0 42                 CALL    MOD
403E D1                       POP     DE
403F 18 CE                    JR      EVAL11
4041                  ;
4041 CD 65 40         MINS:   CALL    EVAL19
4044 7C                       LD      A,H
4045 2F                       CPL
4046 67                       LD      H,A
4047 7D                       LD      A,L
4048 2F                       CPL
4049 6F                       LD      L,A
404A 23                       INC     HL
404B C9                       RET
404C                  ;
404C 4F               CHAR:   LD      C,A
404D 13                       INC     DE
404E 1A                       LD      A,(DE)
404F FE 0D                    CP      CR
4051 28 0F                    JR      Z,CHAR1
4053 FE 5E                    CP      '^'
4055 20 03                    JR      NZ,CHAR0
4057 CD 51 3F                 CALL    ESC
405A 6F               CHAR0:  LD      L,A
405B 26 00                    LD      H,0
405D 13                       INC     DE
405E 1A                       LD      A,(DE)
405F 13                       INC     DE
4060 B9                       CP      C
4061 C8                       RET     Z
4062 C3 2E 43         CHAR1:  JP      ERR12
4065                  ;
4065 13               EVAL19: INC     DE
4066 CD A6 3F         EVAL20: CALL    SPCUT
4069 FE 2D                    CP      '-'
406B 28 D4                    JR      Z,MINS
406D FE 24                    CP      '$'
406F 28 48                    JR      Z,HEX
4071 FE 25                    CP      '%'
4073 28 61                    JR      Z,BIN
4075 CD 9E 3F                 CALL    ISDEC
4078 30 70                    JR      NC,DECI
407A C6 30                    ADD     A,'0'
407C CD E0 3F                 CALL    ISQ
407F 28 CB                    JR      Z,CHAR
4081 21 C7 3F                 LD      HL,ESPTBL
4084 01 0C 00                 LD      BC,SPREND-ESPTBL
4087 ED B1                    CPIR
4089 CA 25 43                 JP      Z,ERR9
408C 08               LABEL:  EX      AF,AF'
408D F5                       PUSH    AF
408E D5                       PUSH    DE
408F CD F8 36                 CALL    HASH
4092 E5                       PUSH    HL               ;HL=hash
4093 D9                       EXX
4094 E1                       POP     HL               ;hash
4095 D1                       POP     DE               ;strptr
4096 CD B2 36                 CALL    ISDEF
4099 30 0D                    JR      NC,LABEL0
409B 3A EB 45                 LD      A,(PASS)
409E B7                       OR      A                ;pass2?
409F C2 0D 43                 JP      NZ,ERR1          ;not def
40A2 21 00 00                 LD      HL,0
40A5 E5                       PUSH    HL
40A6 18 0C                    JR      LABEL1
40A8 69               LABEL0: LD      L,C
40A9 60                       LD      H,B
40AA CD 94 1F                 CALL    #PEEK
40AD 4F                       LD      C,A
40AE 23                       INC     HL
40AF CD 94 1F                 CALL    #PEEK
40B2 47                       LD      B,A
40B3 C5                       PUSH    BC
40B4 D9               LABEL1: EXX
40B5 E1                       POP     HL
40B6 F1                       POP     AF
40B7 08                       EX      AF,AF'           ;F'=flags
40B8 C9                       RET
40B9                  ;
40B9 13               HEX:    INC     DE
40BA 1A                       LD      A,(DE)
40BB CD B8 1F                 CALL    #HEX
40BE 30 04                    JR      NC,HEX0
40C0 2A D0 45                 LD      HL,(LOGADR)
40C3 C9                       RET
40C4 6F               HEX0:   LD      L,A
40C5 26 00                    LD      H,0
40C7 13                       INC     DE
40C8 1A               HEX1:   LD      A,(DE)
40C9 CD B8 1F                 CALL    #HEX
40CC D8                       RET     C
40CD 29                       ADD     HL,HL
40CE 29                       ADD     HL,HL
40CF 29                       ADD     HL,HL
40D0 29                       ADD     HL,HL
40D1 B5                       OR      L
40D2 6F                       LD      L,A
40D3 13                       INC     DE
40D4 18 F2                    JR      HEX1
40D6                  ;
40D6 21 00 00         BIN:    LD      HL,0
40D9 13               BIN0:   INC     DE
40DA 1A                       LD      A,(DE)
40DB FE 5F                    CP      '_'
40DD 28 FA                    JR      Z,BIN0
40DF D6 30            ISBIN:  SUB     '0'
40E1 D8                       RET     C
40E2 FE 02                    CP      2
40E4 D0                       RET     NC
40E5 0F                       RRCA
40E6 ED 6A                    ADC     HL,HL
40E8 18 EF                    JR      BIN0
40EA                  ;
40EA D5               DECI:   PUSH    DE
40EB 4F                       LD      C,A
40EC 13               DECI0:  INC     DE
40ED CD B9 3F                 CALL    ISLSPR0
40F0 20 FA                    JR      NZ,DECI0
40F2 1B                       DEC     DE
40F3 1A                       LD      A,(DE)
40F4 D1                       POP     DE
40F5 FE 48                    CP      'H'
40F7 20 0C                    JR      NZ,DECI1
40F9 79                       LD      A,C
40FA CD C4 40                 CALL    HEX0
40FD 1A                       LD      A,(DE)
40FE FE 48                    CP      'H'
4100 C2 25 43         ERR90:  JP      NZ,ERR9
4103 13                       INC     DE
4104 C9                       RET
4105 FE 42            DECI1:  CP      'B'
4107 20 11                    JR      NZ,DECI2
4109 79                       LD      A,C
410A FE 02                    CP      2
410C D2 25 43                 JP      NC,ERR9
410F 6F                       LD      L,A
4110 26 00                    LD      H,0
4112 CD D9 40                 CALL    BIN0
4115 1A                       LD      A,(DE)
4116 FE 42                    CP      'B'
4118 18 E6                    JR      ERR90
411A 69               DECI2:  LD      L,C
411B 26 00                    LD      H,0
411D 13               DECI3:  INC     DE
411E CD 9D 3F                 CALL    ISDEC0
4121 D8                       RET     C
4122 4D                       LD      C,L
4123 44                       LD      B,H
4124 29                       ADD     HL,HL
4125 29                       ADD     HL,HL
4126 09                       ADD     HL,BC
4127 29                       ADD     HL,HL
4128 4F                       LD      C,A
4129 06 00                    LD      B,0
412B 09                       ADD     HL,BC
412C 18 EF                    JR      DECI3
412E                  ;
412E                  ;       get condition code no
412E                  ;
412E 21 9B 41         GETCND: LD      HL,CNDTBL
4131 06 08                    LD      B,8
4133 18 13                    JR      GTREG0
4135                  ;
```

▶ひさびさにゲーセンに入ってみて，アフターバーナーIIというのをやってみた。「FIRE, FIRE, FIRE, ズガン」。えっ！　もうやられたの？　難しくてついていけないな，これには。あーあ，ワープ&ワープの時代が懐かしい。　中本　悟（17）長崎県

```
4135 CD 3C 41       REGCHK: CALL    GETREG
4138 D0                     RET     NC
4139 C3 19 43       ERR56:  JP      ERR5
413C                ;
413C                ;       get register no
413C                ;
413C 21 6A 41       GETREG: LD      HL,REGTBL
413F 06 11                  LD      B,17
4141 18 05                  JR      GTREG0
4143                ;
4143                ;       get 8bit register no
4143                ;
4143 21 79 41       GETRG8: LD      HL,RG8TBL
4146 06 08                  LD      B,8
4148 C5             GTREG0: PUSH    BC
4149 48                     LD      C,B
414A D5             GTREG1: PUSH    DE
414B 1A             GTREG2: LD      A,(DE)
414C BE                     CP      (HL)
414D 20 04                  JR      NZ,GTREG3
414F 13                     INC     DE
4150 23                     INC     HL
4151 18 F8                  JR      GTREG2
4153 34             GTREG3: INC     (HL)
4154 35                     DEC     (HL)
4155 20 0A                  JR      NZ,GTREG4
4157 CD BA 3F               CALL    ISLSPR
415A 20 05                  JR      NZ,GTREG4
415C E1                     POP     HL
415D 79                     LD      A,C
415E 90                     SUB     B
415F C1                     POP     BC
4160 C9                     RET                     ;NC
4161 CD 56 42       GTREG4: CALL    SEAZ
4164 D1                     POP     DE
4165 10 E3                  DJNZ    GTREG1
4167 C1                     POP     BC
4168 37                     SCF
4169 C9                     RET
416A                ;
416A 42 43 00       REGTBL: DEFB    'BC',0          ;00
416D 44 45 00               DEFB    'DE',0          ;01
4170 48 4C 00               DEFB    'HL',0          ;02
4173 53 50 00               DEFB    'SP',0          ;03
4176 41 46 00               DEFB    'AF',0          ;04
4179 42 00          RG8TBL: DEFB    'B',0           ;05     00
417B 43 00                  DEFB    'C',0           ;06     01
417D 44 00                  DEFB    'D',0           ;07     02
417F 45 00                  DEFB    'E',0           ;08     03
4181 48 00                  DEFB    'H',0           ;09     04
4183 4C 00                  DEFB    'L',0           ;0A     05
4185 28 48 4C 29            DEFB    '(HL)',0        ;0B     06
4189 00
418A 41 00                  DEFB    'A',0           ;0C     07
418C 49 58 00               DEFB    'IX',0          ;0D
418F 49 59 00               DEFB    'IY',0          ;0E
4192 28 49 58 00            DEFB    '(IX',0         ;0F
4196 28 49 59 00            DEFB    '(IY',0         ;10
419A 00                     DEFB    0
419B                ;
419B 4E 5A 00       CNDTBL: DEFB    'NZ',0          ;00
419E 5A 00                  DEFB    'Z',0           ;01
41A0 4E 43 00               DEFB    'NC',0          ;02
41A3 43 00                  DEFB    'C',0           ;03
41A5 50 4F 00               DEFB    'PO',0          ;04
41A8 50 45 00               DEFB    'PE',0          ;05
41AB 50 00                  DEFB    'P',0           ;06
41AD 4D 00                  DEFB    'M',0           ;07
41AF 00                     DEFB    0
41B0                ;
41B0 34             CMPGEN: INC     (HL)
41B1 35                     DEC     (HL)
41B2 37                     SCF
41B3 C8                     RET     Z
41B4 4B                     LD      C,E
41B5 42                     LD      B,D
41B6 1A             CPGEN0: LD      A,(DE)
41B7 BE                     CP      (HL)
41B8 20 05                  JR      NZ,CPGEN1
41BA 13                     INC     DE
41BB 23                     INC     HL
41BC C3 B6 41               JP      CPGEN0
41BF 34             CPGEN1: INC     (HL)
41C0 35                     DEC     (HL)
41C1 23                     INC     HL
41C2 20 05                  JR      NZ,CPGEN2
41C4 CD D4 3F               CALL    ISBLNK
41C7 28 0A                  JR      Z,CPGEN3
41C9 CD 56 42       CPGEN2: CALL    SEAZ
41CC 23                     INC     HL
41CD 23                     INC     HL
41CE 59                     LD      E,C
41CF 50                     LD      D,B
41D0 C3 B0 41               JP      CMPGEN
41D3 4E             CPGEN3: LD      C,(HL)
41D4 23                     INC     HL
41D5 46                     LD      B,(HL)
41D6 78                     LD      A,B
41D7 B7                     OR      A
41D8 20 05                  JR      NZ,CPGEN4
41DA CD 67 37               CALL    PUTC
41DD B7                     OR      A
41DE C9                     RET
41DF CD A2 37       CPGEN4: CALL    PUT2
41E2 B7                     OR      A
41E3 C9                     RET
41E4                ;
41E4 28 44 45 29    []TBL:  DEFB    '(DE),A',0
41E8 2C 41 00
41EB 12                     LD      (DE),A
41EC 00                     NOP
41ED 28 42 43 29            DEFB    '(BC),A',0
41F1 2C 41 00
41F4 02                     LD      (BC),A
41F5 00                     NOP
41F6 52 2C 41 00            DEFB    'R,A',0
41FA ED 4F                  LD      R,A
41FC 49 2C 41 00            DEFB    'I,A',0
4200 ED 47                  LD      I,A
4202 00                     DEFB    0
4203                ;
4203 44 45 29 00    A[]TBL: DEFB    'DE)',0
4207 1A                     LD      A,(DE)
4208 00                     NOP
4209 42 43 29 00            DEFB    'BC)',0
420D 0A                     LD      A,(BC)
420E 00                     NOP
420F 00                     DEFB    0
4210                ;
4210 44 45 2C 48    EXTBL:  DEFB    'DE,HL',0
4214 4C 00
4216 EB                     EX      DE,HL
4217 00                     NOP
4218 41 46 2C 41            DEFB    'AF,AF'',0
421C 46 27 00
421F 08                     EX      AF,AF'
4220 00                     NOP
4221 28 53 50 29            DEFB    '(SP),HL',0
4225 2C 48 4C 00
4229 E3                     EX      (SP),HL
422A 00                     NOP
422B 28 53 50 29            DEFB    '(SP),IX',0
422F 2C 49 58 00
4233 DD E3                  EX      (SP),IX
4235 28 53 50 29            DEFB    '(SP),IY',0
4239 2C 49 59 00
423D FD E3                  EX      (SP),IY
423F 00                     DEFB    0
4240                ;
4240 28 48 4C 29    JPTBL:  DEFB    '(HL)',0
4244 00
4245 E9                     JP      (HL)
4246 00                     NOP
4247 28 49 58 29            DEFB    '(IX)',0
424B 00
424C DD E9                  JP      (IX)
424E 28 49 59 29            DEFB    '(IY)',0
4252 00
4253 FD E9                  JP      (IY)
4255 00                     DEFB    0
4256                ;
4256                ;       skip to next eos
4256                ;
4256 AF             SEAZ:   XOR     A
4257 BE             SEAZ0:  CP      (HL)
4258 23                     INC     HL
4259 C8                     RET     Z
425A C3 57 42               JP      SEAZ0
425D                ;
425D                ;       print in decimal
425D                ;
425D D5             PRDEC:  PUSH    DE
425E F5                     PUSH    AF
425F 01 0A 05               LD      BC,050AH
4262 11 C5 45               LD      DE,DECWK+5
4265 CD A0 42       PRDEC0: CALL    DIVC
4268 C6 30                  ADD     A,'0'
426A 1B                     DEC     DE
426B 12                     LD      (DE),A
426C 10 F7                  DJNZ    PRDEC0
426E 6B                     LD      L,E
426F 62                     LD      H,D
4270 01 20 04               LD      BC,0420H
4273 3E 30                  LD      A,'0'
4275 BE             PRDEC1: CP      (HL)
4276 20 04                  JR      NZ,PRDEC2
4278 71                     LD      (HL),C
4279 23                     INC     HL
427A 10 F9                  DJNZ    PRDEC1
427C F1             PRDEC2: POP     AF
427D B7                     OR      A
427E 28 01                  JR      Z,PRDEC3
4280 EB                     EX      DE,HL
4281 CD E5 1F       PRDEC3: CALL    #MSX
4284 D1                     POP     DE
4285 C9                     RET
4286                ;
4286                ;       HL*BC=HL
4286                ;
4286 D5             MUL:    PUSH    DE
4287 EB                     EX      DE,HL
4288 21 00 00               LD      HL,0
428B 3E 10                  LD      A,16
428D 29             MUL0:   ADD     HL,HL
428E EB                     EX      DE,HL
428F 29                     ADD     HL,HL
4290 EB                     EX      DE,HL
4291 30 01                  JR      NC,MUL1
4293 09                     ADD     HL,BC
4294 3D             MUL1:   DEC     A
4295 20 F6                  JR      NZ,MUL0
4297 D1                     POP     DE
4298 C9                     RET
4299                ;
4299                ;       HL/BC=HL...DE
4299                ;
4299 D5             DIV:    PUSH    DE
429A CD B0 42               CALL    MOD
429D EB                     EX      DE,HL
429E D1                     POP     DE
429F C9                     RET
42A0                ;
42A0                ;       HL/C=HL...A
42A0                ;
42A0 C5             DIVC:   PUSH    BC
42A1 AF                     XOR     A
42A2 06 10                  LD      B,16
42A4 29             DIVC0:  ADD     HL,HL
42A5 17                     RLA
42A6 2C                     INC     L
42A7 91                     SUB     C
42A8 30 02                  JR      NC,DIVC1
42AA 2D                     DEC     L
42AB 81                     ADD     A,C
42AC 10 F6          DIVC1:  DJNZ    DIVC0
42AE C1                     POP     BC
42AF C9                     RET
42B0                ;
42B0                ;       HL/BC=DE...HL
42B0                ;
42B0 78             MOD:    LD      A,B
42B1 B1                     OR      C
42B2 28 71                  JR      Z,ERR9
42B4 EB                     EX      DE,HL
42B5 21 00 00               LD      HL,0
42B8 3E 10                  LD      A,16
42BA EB             MOD0:   EX      DE,HL
42BB 29                     ADD     HL,HL
42BC EB                     EX      DE,HL
42BD ED 6A                  ADC     HL,HL
42BF 1C                     INC     E
42C0 ED 42                  SBC     HL,BC
42C2 30 02                  JR      NC,MOD1
42C4 09                     ADD     HL,BC
42C5 1D                     DEC     E
42C6 3D             MOD1:   DEC     A
42C7 20 F1                  JR      NZ,MOD0
42C9 C9                     RET
42CA                ;
42CA                ;       get displacement
42CA                ;               of index addressing
42CA                ;
42CA CD E2 42       GETXD0: CALL    EVAL]
42CD AF                     XOR     A
42CE 91                     SUB     C
42CF 4F                     LD      C,A
42D0 C9                     RET
42D1 CD 80 37       PUT[X]D:CALL    PUT[X]
42D4 1A             GETIXD: LD      A,(DE)
42D5 13                     INC     DE
42D6 D6 29                  SUB     ')'
42D8 4F                     LD      C,A
42D9 C8                     RET     Z
42DA FE 04                  CP      '-'-')'
42DC 28 EC                  JR      Z,GETXD0
42DE FE 02                  CP      '+'-')'
42E0 20 37                  JR      NZ,ERR5
42E2                ;
42E2                ;       evaluate 'expression)'
42E2                ;
42E2 CD E6 3F       EVAL]:  CALL    EVAL
42E5                        ;
42E5 1A             ]CHK:   LD      A,(DE)
42E6 FE 29                  CP      ')'
42E8 20 3E                  JR      NZ,ERR10
42EA 13                     INC     DE
42EB C9                     RET
42EC                ;
42EC                ;       evaluate '(expression)'
42EC                ;
42EC CD FC 42       [EVAL]: CALL    IS[
42EF 20 28                  JR      NZ,ERR5
42F1 18 EF                  JR      EVAL]
42F3                ;
42F3                ;       check ','
42F3                ;
42F3 47             CMACHK: LD      B,A
42F4 1A                     LD      A,(DE)
42F5 FE 2C                  CP      ','
42F7 78                     LD      A,B
42F8 20 31                  JR      NZ,ERR11
42FA 13                     INC     DE
42FB C9                     RET
42FC                ;
42FC                ;       check '['
42FC                ;
42FC 1A             IS[:    LD      A,(DE)
42FD FE 28                  CP      '('
42FF C0                     RET     NZ
4300 13                     INC     DE
4301 C9                     RET
4302                ;
```


▶STUDIO Xに載っていた田中さん（システムハウス勤務）という方は，ソフトハウスの結婚ブームのなかで，結婚が決まったとログインに書いてあった人だと思う。なんとなく2つの雑誌に載っている人を見ると，嬉しくなってしまう小市民。

森　幹雄　(19)　愛知県

```
4302                ;       check whether line pointer is
4302                ;                              at end of line
4302                ;
4302 CD A6 3F       EOLCHK: CALL    SPCUT
4305 FE 0D                  CP      CR
4307 C8                     RET     Z
4308 FE 3B                  CP      ';'
430A C8                     RET     Z
430B                ;
430B                ;       error
430B                ;
430B AF             ERR0:   XOR     A
430C 01                     DEFB    01H
430D 3E 01          ERR1:   LD      A,1
430F 01                     DEFB    01H
4310 3E 02          ERR2:   LD      A,2
4312 01                     DEFB    01H
4313 3E 03          ERR3:   LD      A,3
4315 01                     DEFB    01H
4316 3E 04          ERR4:   LD      A,4
4318 01                     DEFB    01H
4319 3E 05          ERR5:   LD      A,5
431B 01                     DEFB    01H
431C 3E 06          ERR6:   LD      A,6
431E 01                     DEFB    01H
431F 3E 07          ERR7:   LD      A,7
4321 01                     DEFB    01H
4322 3E 08          ERR8:   LD      A,8
4324 01                     DEFB    01H
4325 3E 09          ERR9:   LD      A,9
4327 01                     DEFB    01H
4328 3E 0A          ERR10:  LD      A,10
432A 01                     DEFB    01H
432B 3E 0B          ERR11:  LD      A,11
432D 01                     DEFB    01H
432E 3E 0C          ERR12:  LD      A,12
4330 01                     DEFB    01H
4331 3E 0D          ERR13:  LD      A,13
4333 01                     DEFB    01H
4334 3E 0E          ERR14:  LD      A,14
4336 01                     DEFB    01H
4337 3E 0F          ERR15:  LD      A,15
4339 ED 53 06 30    ERROR:  LD      (ERRPTR),DE
433D 87                     ADD     A,A
433E 21 20 45               LD      HL,ERRTBL
4341 4F                     LD      C,A
4342 06 00                  LD      B,0
4344 09                     ADD     HL,BC
4345 5E                     LD      E,(HL)
4346 23                     INC     HL
4347 56                     LD      D,(HL)
4348 3A F4 45               LD      A,(EEDFLG)
434B B7                     OR      A
434C 20 29                  JR      NZ,REEDIT
434E CD EE 1F               CALL    #LTNL
4351 CD 72 44               CALL    @MSX
4354 CD E2 1F               CALL    #MPRNT
4357 20 45 72 72            DEFB    ' Error',CR,0
435B 6F 72 0D 00
435F 3A EB 45               LD      A,(PASS)
4362 3C                     INC     A
4363 28 0F                  JR      Z,ERROR0
4365 CD B6 43               CALL    PRTXT
4368 3A EA 45               LD      A,(ERRPOS)
436B 47                     LD      B,A
436C CD DF 1F               CALL    #TAB
436F 3E 5E                  LD      A,'^^'
4371 CD F4 1F               CALL    #PRINT
4374 C3 2B 30       ERROR0: JP      HOT0
4377                ;
4377                ;       branch to editor at error
4377                ;
4377 ED 7B C6 45    REEDIT: LD      SP,(SPBUF)
437B 21 2B 30               LD      HL,HOT0
437E E5                     PUSH    HL
437F C3 1D 46               JP      EDERR
4382                ;
4382                ;       print list
4382                ;
4382 D5             PRLST:  PUSH    DE
4383 CD 43 44               CALL    LPTON
4386 2A E6 45               LD      HL,(PADR)
4389 CD BE 1F               CALL    #PRTHL
438C 3A D4 45               LD      A,(LBLFLG)
438F FE 02                  CP      2
4391 20 09                  JR      NZ,PRLST0
4393 CD F1 1F               CALL    #PRNTS
4396 3E 50                  LD      A,'P'
4398 CD F4 1F               CALL    #PRINT
439B B7                     OR      A
439C C4 10 44       PRLST0: CALL    NZ,PROBJ
439F 06 10                  LD      B,16
43A1 CD DF 1F               CALL    #TAB
43A4 CD B6 43               CALL    PRTXT
43A7 CD FA 43       PRLST1: CALL    PROBJ2
43AA 20 FB                  JR      NZ,PRLST1
43AC CD D6 1F               CALL    #LPTOF
43AF CD C7 1F               CALL    #PAUSE
43B2 7A 33                  DEFW    ABT
43B4 D1                     POP     DE
43B5 C9                     RET
43B6                ;
43B6                ;       print a line of text
43B6                ;
43B6 D5             PRTXT:  PUSH    DE
43B7 CD F1 1F               CALL    #PRNTS
43BA 2A CA 45               LD      HL,(LINNO)
43BD AF                     XOR     A
43BE CD 5D 42               CALL    PRDEC
43C1 CD F1 1F               CALL    #PRNTS
43C4 CD F1 1F               CALL    #PRNTS
43C7 ED 5B C8 45            LD      DE,(LINPTR)
43CB 2A 06 30       PRTXT0: LD      HL,(ERRPTR)
43CE B7                     OR      A
43CF ED 52                  SBC     HL,DE
43D1 20 07                  JR      NZ,PRTXT1
43D3 CD 18 20               CALL    #CSR
43D6 7D                     LD      A,L
43D7 32 EA 45               LD      (ERRPOS),A
43DA 1A             PRTXT1: LD      A,(DE)
43DB 13                     INC     DE
43DC FE 0D                  CP      CR
43DE 28 17                  JR      Z,PRTXT3
43E0 FE 09                  CP      TAB
43E2 20 0E                  JR      NZ,PRTXT2
43E4 CD 18 20               CALL    #CSR
43E7 3E 08                  LD      A,8
43E9 85                     ADD     A,L
43EA E6 F8                  AND     0F8H
43EC 47                     LD      B,A
43ED CD B5 32               CALL    WCHK
43F0 18 D9                  JR      PRTXT0
43F2 CD F4 1F       PRTXT2: CALL    #PRINT
43F5 18 D4                  JR      PRTXT0
43F7 D1             PRTXT3: POP     DE
43F8 18 13                  JR      @@NL
43FA                ;
43FA                ;       print objext code
43FA                ;
43FA CD 2A 44       PROBJ2: CALL    OBJC
43FD C8                     RET     Z
43FE E5                     PUSH    HL
43FF ED 4B CE 45            LD      BC,(OFSADR)
4403 B7                     OR      A
4404 ED 42                  SBC     HL,BC
4406 CD BE 1F               CALL    #PRTHL
4409 E1                     POP     HL
440A CD 14 44               CALL    PROB0
440D C3 EE 1F       @@NL:   JP      #LTNL
4410                ;
4410 CD 2A 44       PROBJ:  CALL    OBJC
4413 C8                     RET     Z
4414 06 04          PROB0:  LD      B,4
4416 7B             PROB1:  LD      A,E
4417 B2                     OR      D
4418 C8                     RET     Z
4419 1B                     DEC     DE
441A 7E                     LD      A,(HL)
441B 23                     INC     HL
441C 22 E8 45               LD      (LOBPTR),HL
441F CD F1 1F               CALL    #PRNTS
4422 CD C1 1F               CALL    #PRTHX
4425 10 EF                  DJNZ    PROB1
4427 AF                     XOR     A
4428 3D                     DEC     A
4429 C9                     RET                     ;NZ
442A                ;
442A DD E5          OBJC:   PUSH    IX
442C E1                     POP     HL
442D ED 5B E8 45            LD      DE,(LOBPTR)
4431 D5                     PUSH    DE
4432 B7                     OR      A
4433 ED 52                  SBC     HL,DE
4435 EB                     EX      DE,HL
4436 E1                     POP     HL
4437 C9                     RET
4438                ;
4438                ;       print filename
4438                ;
4438 3A 5D 1F       FPRNT:  LD      A,(#DSK)
443B CD 51 31               CALL    PRDEV
443E CD 9D 1F               CALL    #FPRNT
4441 18 CA                  JR      @@NL
4443                ;
4443                ;       printer switch on
4443                ;
4443 3A 11 30       LPTON:  LD      A,(LPTSW)
4446 B7                     OR      A
4447 C8                     RET     Z
4448 CD D9 1F               CALL    #LPTON
444B AF                     XOR     A
444C CD DC 1F               CALL    #LPRNT
444F D0                     RET     NC
4450 AF                     XOR     A
4451 32 11 30               LD      (LPTSW),A
4454 CD D6 1F               CALL    #LPTOF
4457 3E 02                  LD      A,2             ;Device Offline
4459 C3 33 20               JP      #ERROR
445C                ;
445C                ;       check whether device is tape
445C                ;
445C FE 51          ISTAPQ: CP      'Q'
445E C8                     RET     Z
445F FE 53          ISTAPE: CP      'S'
4461 C8                     RET     Z
4462 FE 54                  CP      'T'
4464 C9                     RET
4465                ;
4465                ;       print string
4465                ;
4465 E3             @MPRNT: EX      (SP),HL
4466 7E             @MPRT0: LD      A,(HL)
4467 23                     INC     HL
4468 B7                     OR      A
4469 28 05                  JR      Z,@MPRT1
446B CD 81 44               CALL    @PRINT
446E 18 F6                  JR      @MPRT0
4470 E3             @MPRT1: EX      (SP),HL
4471 C9                     RET
4472                ;
4472 F5             @MSX:   PUSH    AF
4473 D5                     PUSH    DE
4474 1A             @MSX0:  LD      A,(DE)
4475 13                     INC     DE
4476 B7                     OR      A
4477 28 05                  JR      Z,@MSX1
4479 CD 81 44               CALL    @PRINT
447C 18 F6                  JR      @MSX0
447E D1             @MSX1:  POP     DE
447F F1                     POP     AF
4480 C9                     RET
4481                ;
4481 B7             @PRINT: OR      A
4482 F2 F4 1F               JP      P,#PRINT
4485 D5                     PUSH    DE
4486 E5                     PUSH    HL
4487 87                     ADD     A,A
4488 6F                     LD      L,A
4489 26 00                  LD      H,0
448B 11 98 44               LD      DE,DICTBL
448E 19                     ADD     HL,DE
448F 5E                     LD      E,(HL)
4490 23                     INC     HL
4491 56                     LD      D,(HL)
4492 CD E5 1F               CALL    #MSX
4495 E1                     POP     HL
4496 D1                     POP     DE
4497 C9                     RET
4498                ;
4498 51 45 B2 44    DICTBL: DEFW    @LBLMS,@MISMS
449C BB 44 EF 44            DEFW    @ILLMS,@CHKMS
44A0 B9 45 C4 44            DEFW    @MEMMS,@TABMS
44A4 05 45 CB 44            DEFW    @ASMMS,SOUMES
44A8 D3 44 0A 32            DEFW    OBJMES,@AREMS
44AC E9 44 95 34            DEFW    @SAVMS,@INGMS
44B0 50 45                  DEFW    @SLBMS
44B2                ;
44B2 4D 69 73 73    @MISMS: DEFB    'Missing ',0
44B6 69 6E 67 20
44BA 00
44BB 49 6C 6C 65    @ILLMS: DEFB    'Illegal ',0
44BF 67 61 6C 20
44C3 00
44C4 20 54 61 62    @TABMS: DEFB    ' Table',0
44C8 6C 65 00
44CB 53 6F 75 72    SOUMES: DEFB    'Source ',0
44CF 63 65 20 00
44D3 4F 62 6A 65    OBJMES: DEFB    'Object ',0
44D7 63 74 20 00
44DB 52 65 77 69    REWMES: DEFB    'Rewind ',0
44DF 6E 64 20 00
44E3                ;
44E3 4B 61 6E 6A    SWMES:  DEFB    'Kanji',0
44E7 69 00
44E9 53 61 76 65    @SAVMS: DEFB    'Save',0
44ED 00
44EE 84                     DEFB    @MEM
44EF 20 43 68 65    @CHKMS: DEFB    ' Check',0
44F3 63 6B 00
44F6 80 85 83 00            DEFB    @LBL,@TAB,@CHK,0
44FA 46 72 61 63            DEFB    'Fractional '
44FE 74 69 6F 6E
4502 61 6C 20
4505 41 73 73 65    @ASMMS: DEFB    'Assemble',0
4509 6D 62 6C 65
450D 00
450E 45 4F 4C 83            DEFB    'EOL',@CHK,0
4512 00
4513 00                     DEFB    0                ;dummy
4514 43 6C 65 61            DEFB    'Clear Space',0
4518 72 20 53 70
451C 61 63 65 00
4520                ;
4520                ;       error messages
4520                ;
4520 40 45 47 45    ERRTBL: DEFW    ERMS00,ERMS01
4524 57 45 64 45            DEFW    ERMS02,ERMS03
4528 67 45 70 45            DEFW    ERMS04,ERMS05
452C 79 45 84 45            DEFW    ERMS06,ERMS07
4530 87 45 8F 45            DEFW    ERMS08,ERMS09
4534 9B 45 A0 45            DEFW    ERMS10,ERMS11
4538 A5 45 AC 45            DEFW    ERMS12,ERMS13
453C B4 45 B9 45            DEFW    ERMS14,ERMS15
4540                ;
4540 53 79 6E 74    ERMS00: DEFB    'Syntax',0
4544 61 78 00
4547 55 6E 64 65    ERMS01: DEFB    'Undefined'
454B 66 69 6E 65
454F 64
4550 20             @SLBMS: DEFB    ' '
4551 4C 61 62 65    @LBLMS: DEFB    'Label',0
4555 6C 00
```

▶ビデオデッキに続いてミニコンポを買ってしまい，楽しんでいます。やっぱりいい音が出ますね。ついでにFM音源も接続して改めて聞いてみましたが，これまたいい音で思わず感動してしまいました。これまで入力していなかった曲も聞いてみたくなりました。

沓掛　浩　(21)　長野県

```
4557 52 65 64 65  ERMS02: DEFB    'Redefinition',0
455B 66 69 6E 69
455F 74 69 6F 6E
4563 00
4564 82 80 00     ERMS03: DEFB    @ILL,@LBL,0
4567 82 4F 70 65  ERMS04: DEFB    @ILL,'Opecode',0
456B 63 6F 64 65
456F 00
4570 82 4F 70 65  ERMS05: DEFB    @ILL,'Operand',0
4574 72 61 6E 64
4578 00
4579 54 6F 6F 20  ERMS06: DEFB    'Too Many',@SLB,'s',0
457D 4D 61 6E 79
4581 8C 73 00
4584 81 80 00     ERMS07: DEFB    @MIS,@LBL,0
4587 54 6F 6F 20  ERMS08: DEFB    'Too Far',0
458B 46 61 72 00
458F 82 45 78 70  ERMS09: DEFB    @ILL,'Expression',0
4593 72 65 73 73
4597 69 6F 6E 00
459B 81 5B 29 5D  ERMS10: DEFB    @MIS,'[)]',0
459F 00
45A0 81 5B 2C 5D  ERMS11: DEFB    @MIS,'[,]',0
45A4 00
45A5 81 51 75 6F  ERMS12: DEFB    @MIS,'Quote',0
45A9 74 65 00
45AC 82 4F 46 46  ERMS13: DEFB    @ILL,'OFFSET',0
45B0 53 45 54 00
45B4 82 4F 52 47  ERMS14: DEFB    @ILL,'ORG',0
45B8 00
45B9              ERMS15:
45B9 4D 65 6D 6F  @MEMMS: DEFB    'Memory',0
45BD 72 79 00
45C0              ;
45C0              ;       works
45C0              ;
45C0 00 00 00 00  DECWK:  DEFW    0,0,0   ;work for PRDEC
```

```
45C4 00 00
45C6              SPBUF:  DEFS    2       ;sp reserve buffer
45C8              LINPTR: DEFS    2       ;current line pointer
45CA              LINNO:  DEFS    2       ;current line no
45CC              ORGADR: DEFS    2       ;origin address
45CE              OFSADR: DEFS    2       ;offset address
45D0              LOGADR: DEFS    2       ;current logical address
45D2              ORGFLG: DEFS    1       ;whether defined ORG
45D3              OFSFLG: DEFS    1       ;whether defined OFFSET
45D4              LBLFLG: DEFS    1       ;whether label defined
45D5                                      ;       in the line or not
45D5              LSTLBL: DEFS    2       ;position of last defined label
45D7              HSHSIZ: DEFS    2       ;size of hash table
45D9              LBLMAX: DEFS    2       ;max of label
45DB              LBLPTR: DEFS    2       ;current label pointer
45DD              LBLCNT: DEFS    2       ;counter of defined label
45DF              HSHMSK: DEFS    1       ;mask of hash
45E0              WKMAX:  DEFS    2       ;end of label table
45E2              SAVADR: DEFS    2       ;object start address for save
45E4              OBJST:  DEFS    2       ;object start address
45E6              PADR:   DEFS    2       ;address for print out
45E8              LOBPTR: DEFS    2       ;object address
45EA                                      ;       at top of the line
45EA              ERRPOS: DEFS    1       ;error occurred x-coodinate
45EB              PASS:   DEFS    1       ;current pass counter
45EC              FLCNT:  DEFS    1       ;counter of source files
45ED              FILPT1: DEFS    2       ;pointer to 1st filename
45EF              FILPTR: DEFS    2       ;current pointer to filename
45F1              LSTSW:  DEFS    1       ;listing on/off switch1
45F2              LSTSW2: DEFS    1       ;listing on/off switch2
45F3              LSTSW3: DEFS    1       ;listing on/off switch3
45F4              EEDFLG: DEFS    1       ;whether branch
45F5                                      ;       to editor at error
45F5              ENDofASM:
```

リスト3 エディタソースリスト

```
0000              ;
0000              ; SMALL Editor for S-OS "SWORD"
0000              ;
0000                      OFFSET  9600H-4617H
4617                      ORG     4617H
4617                      ;
1FF4 P            PRINT:  EQU     1FF4H
1FF1 P            PRNTS:  EQU     1FF1H
1FEE P            LETNL:  EQU     1FEEH
1FE8 P            MSG:    EQU     1FE8H
1FE5 P            MSX:    EQU     1FE5H
1FD3 P            GETL:   EQU     1FD3H
1FD0 P            GETKY:  EQU     1FD0H
1FC4 P            BELL:   EQU     1FC4H
1FBE P            PRTHL:  EQU     1FBEH
1FB5 P            AHEX:   EQU     1FB5H
1FB2 P            HLHEX:  EQU     1FB2H
1FAF P            WOPEN:  EQU     1FAFH
1FAC P            WRD:    EQU     1FACH
1FA6 P            RDD:    EQU     1FA6H
1FA3 P            FILE:   EQU     1FA3H
1F9D P            FPRNT:  EQU     1F9DH
2006 P            DIR:    EQU     2006H
2009 P            ROPEN:  EQU     2009H
2021 P            FLGET:  EQU     2021H
201E P            CSRSET: EQU     201EH
2018 P            CSRRD:  EQU     2018H
2024 P            RDVSW:  EQU     2024H
2033 P            ERROR:  EQU     2033H
1F81 P            [HL]:   EQU     1F81H
4617                      ;
1F76 P            KBFAD:  EQU     1F76H
1F72 P            SIZE:   EQU     1F72H
1F70 P            DTADR:  EQU     1F70H
1F6E P            EXADR:  EQU     1F6EH
1F6A P            MEMAX:  EQU     1F6AH
1F5D P            DSK:    EQU     1F5DH
1F5C P            WIDTH:  EQU     1F5CH
1F5B P            MAXLN:  EQU     1F5BH
2E00 P            BUFFER: EQU     2E00H
3006 P            #ERRAD: EQU     3006H
3008 P            #TXTST: EQU     3008H
300A P            PRER:   EQU     300AH
4617
4617 C3 48 46             JP      COLD
461A C3 64 46             JP      HOT
461D
461D              ERR:
461D 2A 06 30             LD      HL,(#ERRAD)
4620 D5                   PUSH    DE
4621 CD 60 4B             CALL    ERRREP
4624 D1                   POP     DE
4625 CD F4 4F             CALL    CMNDLN
4628 CD 0A 30             CALL    PRER
462B 3E 3F                LD      A,'?'
462D CD F4 1F             CALL    PRINT
4630 CD C4 1F             CALL    BELL
4633 21 91 46             LD      HL,CMND
4636 E5                   PUSH    HL
4637 C3 D3 4B             JP      EDIT1
463A              ;
463A              ;#TXTST: DEFW    4E00H
463A 00 00        #TXTEND:DEFW    0
463C 00           #MAXLN: DEFB    0
463D 00 00        #CURADR:DEFW    0
463F 00 00        #CURLN: DEFW    0
4641 00 00        #NXTADR:DEFW    0
4643 00 00        #NXTLN: DEFW    0
4645 00 00 00     #FSTCH: DEFB    0,0,0
4648
4648              COLD:
4648 DD 2A 08 30          LD      IX,(#TXTST)
464C DD 6E 00             LD      L,(IX)
464F DD 66 01             LD      H,(IX+1)
4652 DD 7E 02             LD      A,(IX+2)
4655 22 45 46             LD      (#FSTCH),HL
4658 32 47 46             LD      (#FSTCH+2),A
465B DD 36 00 0D          LD      (IX),0DH
465F DD 36 01 00          LD      (IX+1),0
4663 C9                   RET
4664
4664              HOT:
4664 3A 5B 1F             LD      A,(MAXLN)
4667 32 3C 46             LD      (#MAXLN),A
466A AF                   XOR     A
466B 2A 08 30             LD      HL,(#TXTST)
466E 4F                   LD      C,A
466F 47                   LD      B,A
4670 ED B1                CPIR
4672 77                   LD      (HL),A          ; 0 to 0 is del buf
4673 22 3A 46             LD      (#TXTEND),HL
4676                      ;
4676 DD 2A 08 30          LD      IX,(#TXTST)
467A DD 22 3D 46          LD      (#CURADR),IX
467E DD 36 FF 0D          LD      (IX-1),0DH
4682 DD 77 FE             LD      (IX-2),A
4685 21 01 00             LD      HL,1
4688 22 3F 46             LD      (#CURLN),HL
468B 22 CE 4C             LD      (MLNNO),HL
468E              ;
468E              ; COMMAND LOOP
468E              ;
468E CD 0C 49             CALL    CURPG
4691              CMND:
4691 21 91 46             LD      HL,CMND         ; LOOP ADRS
4694 E5                   PUSH    HL
4695 CD F4 4F             CALL    CMNDLN
4698 3E 3E                LD      A,'>'
469A CD F4 1F             CALL    PRINT
469D CD 21 20     CMND1:  CALL    FLGET
46A0 E6 DF                AND     0DFH            ; 11011111B
46A2 FE 51                CP      'Q'
46A4 20 02                JR      NZ,CMND2
46A6 E1                   POP     HL
46A7 C9                   RET
46A8
46A8 21 C5 46     CMND2:  LD      HL,COMTBL
46AB 06 0C                LD      B,12
46AD CD B2 46             CALL    JPTBL
46B0 D8                   RET     C
46B1 E9                   JP      (HL)
46B2
46B2              JPTBL:
46B2 BE                   CP      (HL)
46B3 20 09                JR      NZ,JPTBL1
46B5 23                   INC     HL
46B6 F5                   PUSH    AF
46B7 7E                   LD      A,(HL)
46B8 23                   INC     HL
46B9 66                   LD      H,(HL)
46BA 6F                   LD      L,A
46BB F1                   POP     AF
46BC B7                   OR      A
46BD C9                   RET
46BE                      ;
46BE 23           JPTBL1: INC     HL
46BF 23                   INC     HL
46C0 23                   INC     HL
46C1 10 EF                DJNZ    JPTBL
46C3 37                   SCF
46C4 C9                   RET
46C5              ;
46C5              COMTBL:
46C5 47                   DEFB    'G'
46C6 E5 48                DEFW    GOLINE
46C8 56                   DEFB    'V'
46C9 15 49                DEFW    NEXTPG
46CB 5A                   DEFB    'Z'
46CC 53 49                DEFW    PREVPG
46CE 45                   DEFB    'E'
46CF CD 4B                DEFW    EDIT
46D1 4C                   DEFB    'L'
46D2 0C 49                DEFW    CURPG
46D4 4E                   DEFB    'N'
46D5 AD 48                DEFW    COLDST
46D7 52                   DEFB    'R'
46D8 CE 48                DEFW    REVIV
46DA 46                   DEFB    'F'
46DB C1 49                DEFW    FILEIO
46DD 54                   DEFB    'T'
46DE B2 48                DEFW    TXTAD
46E0 53                   DEFB    'S'
46E1 E0 47                DEFW    SEARCH
46E3 43                   DEFB    'C'
46E4 0B 47                DEFW    REPLACE
46E6 4D                   DEFB    'M'
46E7 E9 46                DEFW    REPORT
46E9
46E9              ; MEMORY REPORT
46E9              ;
46E9              REPORT:
46E9 CD F4 4F             CALL    CMNDLN
46EC CD F1 1F             CALL    PRNTS
46EF 2A 08 30             LD      HL,(#TXTST)
46F2 CD BE 1F             CALL    PRTHL
46F5 3E 2D                LD      A,'-'
46F7 F5                   PUSH    AF
46F8 CD F4 1F             CALL    PRINT
46FB CD 33 4E             CALL    SCHEND
46FE CD BE 1F             CALL    PRTHL
4701 F1                   POP     AF
4702 CD F4 1F             CALL    PRINT
4705 2A 3A 46             LD      HL,(#TXTEND)
4708 C3 BE 1F             JP      PRTHL
470B
470B              ; REPLACE FORWARD
470B              ;
470B              REPLACE:
470B CD 2F 48             CALL    SCH1
470E FE 1B                CP      1BH
4710 C8                   RET     Z
4711 11 C2 47             LD      DE,REPMES
4714 CD A9 4F             CALL    GETCMND
4717 CD 78 48             CALL    BREAK?
471A C8                   RET     Z
471B 1A                   LD      A,(DE)
471C B7                   OR      A
471D C8                   RET     Z
471E 21 C9 47             LD      HL,REPSTR
4721 CD 5A 48             CALL    CNVTRNS
```

▶micro Odysseyを毎号読んでいますが，社会現象というか，身の回りのことがいろいろと書いてありいいと思います。いままで最もよかったのが，地下鉄だか，JRだかの人物看板の話です。

中島　邦彦　(17)　東京都

```
4724                         ;
4724 21 96 48                LD      HL,SCHSTR
4727 CD B5 47                CALL    STRLEN
472A E5                      PUSH    HL
472B 21 C9 47                LD      HL,REPSTR
472E CD B5 47                CALL    STRLEN
4731 D1                      POP     DE
4732 B7                      OR      A
4733 ED 52                   SBC     HL,DE
4735 22 DE 47                LD      (XCHGSZ),HL
4738                         ;
4738 CD A4 47        RPLC1:  CALL    ASK
473B 30 0B                   JR      NC,RPLC3
473D CD 21 20        RPLC2:  CALL    FLGET
4740 FE 0D                   CP      0DH
4742 28 0D                   JR      Z,XCHG
4744 FE 1B                   CP      1BH
4746 20 F0                   JR      NZ,RPLC1
4748 CD C4 1F        RPLC3:  CALL    BELL
474B CD 0C 49                CALL    CURPG
474E C3 D3 4B                JP      EDIT1
4751                         ;
4751 2A DE 47        XCHG:   LD      HL,(XCHGSZ)
4754 7C                      LD      A,H
4755 B7                      OR      A
4756 F4 20 4E                CALL    P,SIZECHK
4759 CA A3 4A                JP      Z,MEMERR
475C 3E C9                   LD      A,0C9H
475E 32 62 4C                LD      (INCSZ1),A
4761 32 7C 4C                LD      (DECSZ1),A
4764 2A DE 47                LD      HL,(XCHGSZ)
4767 7D                      LD      A,L
4768 B4                      OR      H
4769 28 1E                   JR      Z,XCHG2
476B E5                      PUSH    HL
476C 7C                      LD      A,H
476D E6 80                   AND     80H
476F 20 0D                   JR      NZ,XCHG1
4771 2A AB 48                LD      HL,(SCHADR)
4774 2B                      DEC     HL
4775 22 AC 4E                LD      (LNADR),HL
4778 E1                      POP     HL
4779 CD 59 4C                CALL    INCSIZE
477C 18 0B                   JR      XCHG2
477E 2A AB 48        XCHG1:  LD      HL,(SCHADR)
4781 2B                      DEC     HL
4782 22 AC 4E                LD      (LNADR),HL
4785 E1                      POP     HL
4786 CD 64 4C                CALL    DECSIZE
4789 3E 18           XCHG2:  LD      A,18H
478B 32 62 4C                LD      (INCSZ1),A
478E 3E 2A                   LD      A,2AH
4790 32 7C 4C                LD      (DECSZ1),A
4793                         ;
4793 21 C9 47                LD      HL,REPSTR
4796 ED 5B AB 48             LD      DE,(SCHADR)
479A 1B                      DEC     DE
479B 7E              XCHG3:  LD      A,(HL)
479C B7                      OR      A
479D 28 99                   JR      Z,RPLC1
479F 23              XCHG4:  INC     HL
47A0 12                      LD      (DE),A
47A1 13                      INC     DE
47A2 18 F7                   JR      XCHG3
47A4                 ;
47A4                 ASK:
47A4                 ;
47A4 CD 33 4E                CALL    SCHEND
47A7 CD FD 47                CALL    SRCH1
47AA FE 1B                   CP      1BH
47AC C8                      RET     Z
47AD 2A A7 4C                LD      HL,(LOC)
47B0 CD 1E 20                CALL    CSRSET
47B3 37                      SCF
47B4 C9                      RET
47B5                 ;
47B5                 STRLEN:
47B5                 ;
47B5 C5                      PUSH    BC
47B6 4D                      LD      C,L
47B7 44                      LD      B,H
47B8 7E              STRLN1: LD      A,(HL)
47B9 23                      INC     HL
47BA B7                      OR      A
47BB 20 FB                   JR      NZ,STRLN1
47BD B7                      OR      A
47BE ED 42                   SBC     HL,BC
47C0 C1                      POP     BC
47C1 C9                      RET
47C2                 ;
47C2 52 65 70 6C     REPMES: DEFB    "Replce:"
47C6 63 65 3A
47C9 20 20 20 20     REPSTR: DEFB    "          "
47CD 20 20 20 20
47D1 20 20
47D3 20 20 20 20             DEFB    "          ",0
47D7 20 20 20 20
47DB 20 20 00
47DE 00 00           XCHGSZ: DEFW    0
47E0
47E0                 ; SEARCH FORWARD
47E0                 ;
47E0                 SEARCH:
47E0 CD 2F 48                CALL    SCH1
47E3 FE 1B                   CP      1BH
47E5 C8                      RET     Z
47E6 CD A4 47        SEARCH1:CALL    ASK
47E9 D2 C4 1F                JP      NC,BELL
47EC CD 21 20        SEARCH2:CALL    FLGET
47EF FE 0D                   CP      0DH
47F1 28 F3                   JR      Z,SEARCH1
47F3 FE 1B                   CP      1BH
47F5 20 F5                   JR      NZ,SEARCH2
47F7 CD C4 1F                CALL    BELL
47FA C3 D3 4B                JP      EDIT1
47FD                 ;
47FD 2A 44 4E        SRCH1:  LD      HL,(ENDAD)
4800 ED 5B AB 48             LD      DE,(SCHADR)
4804 B7                      OR      A
4805 ED 52                   SBC     HL,DE
4807 7D                      LD      A,L
4808 B4                      OR      H
4809 3E 1B                   LD      A,1BH
480B C8                      RET     Z
480C                         ;
480C 4D                      LD      C,L
480D 44                      LD      B,H
480E EB                      EX      DE,HL
480F 11 96 48                LD      DE,SCHSTR
4812 1A              SRCH2:  LD      A,(DE)
4813 ED B1                   CPIR
4815 3E 1B                   LD      A,1BH
4817 C0                      RET     NZ
4818                         ;
4818 22 AB 48                LD      (SCHADR),HL
481B 2B                      DEC     HL
481C 23              SRCH3:  INC     HL
481D 13                      INC     DE
481E 1A                      LD      A,(DE)
481F B7                      OR      A
4820 28 05                   JR      Z,SRCH4
4822 BE                      CP      (HL)
4823 28 F7                   JR      Z,SRCH3
4825 18 D6                   JR      SRCH1
4827 ED 5B AB 48     SRCH4:  LD      DE,(SCHADR)
482B 1B                      DEC     DE
482C C3 64 4B                JP      ADTOLN
482F                 ;
482F                 SCH1:
482F CD 33 4E                CALL    SCHEND
4832 CD F4 4F                CALL    CMNDLN
4835 11 82 48                LD      DE,BCMES
4838 CD E5 1F                CALL    MSX
483B CD 21 20                CALL    FLGET
483E E6 DF                   AND     0DFH            ; 1101.1111B
4840 FE 42                   CP      'B'
4842 2A 3D 46                LD      HL,(#CURADR)
4845 20 03                   JR      NZ,SCH2
4847 2A 08 30                LD      HL,(#TXTST)
484A 22 AB 48        SCH2:   LD      (SCHADR),HL
484D 11 8F 48                LD      DE,SCHMES
4850 CD A9 4F                CALL    GETCMND
4853 CD 78 48                CALL    BREAK?
4856 C8                      RET     Z
4857                         ;
4857 21 96 48                LD      HL,SCHSTR
485A                 ;
485A                 CNVTRNS:
485A                 ;
485A 06 14                   LD      B,20
485C 1A              CNVTRS1:LD      A,(DE)
485D 13                      INC     DE
485E 77                      LD      (HL),A
485F B7                      OR      A
4860 C8                      RET     Z
4861 FE 5E                   CP      '^^'
4863 28 04                   JR      Z,CNVTRS2
4865 23                      INC     HL
4866 10 F4                   DJNZ    CNVTRS1
4868 C9                      RET
4869 1A              CNVTRS2:LD      A,(DE)
486A 13                      INC     DE
486B FE 5E                   CP      '^^'
486D 28 04                   JR      Z,CNVTRS3
486F E6 DF                   AND     0DFH            ; 1101.1111B
4871 D6 40                   SUB     '@'
4873 77              CNVTRS3:LD      (HL),A
4874 23                      INC     HL
4875 10 E5                   DJNZ    CNVTRS1
4877 C9                      RET
4878                 ;
4878                 BREAK?:
4878                 ;
4878 D5                      PUSH    DE
4879 ED 5B 76 1F             LD      DE,(KBFAD)
487D 1A                      LD      A,(DE)
487E D1                      POP     DE
487F FE 1B                   CP      1BH
4881 C9                      RET
4882                 ;
4882 42 65 67 69     BCMES:  DEFB    "Begin, Here ",0
4886 6E 2C 20 48
488A 65 72 65 20
488E 00
488F 53 65 61 72     SCHMES: DEFB    "Search:"
4893 63 68 3A
4896 20 20 20 20     SCHSTR: DEFB    "          "
489A 20 20 20 20
489E 20 20
48A0 20 20 20 20             DEFB    "          ", 0
48A4 20 20 20 20
48A8 20 20 00
48AB                 ;
48AB 00 00           SCHADR: DEFW    0
48AD
48AD                 ; NEW TEXT
48AD                 ;
48AD                 COLDST:
48AD CD 48 46                CALL    COLD
48B0 18 2F                   JR      REVIV1
48B2
48B2                 ; Change TEXT ADR
48B2                 ;
48B2                 TXTAD:
48B2 11 C4 48                LD      DE,ADMES
48B5 CD A9 4F                CALL    GETCMND
48B8 CD B2 1F                CALL    HLHEX
48BB D8                      RET     C
48BC 22 08 30                LD      (#TXTST),HL
48BF CD 48 46                CALL    COLD
48C2 18 1D                   JR      REVIV1
48C4                 ;
48C4 54 65 78 74     ADMES:  DEFB    "Text Adr:", 0
48C8 20 41 64 72
48CC 3A 00
48CE
48CE                 ; Revive Text
48CE                 ;
48CE                 REVIV:
48CE DD 2A 08 30             LD      IX,(#TXTST)
48D2 2A 45 46                LD      HL,(#FSTCH)
48D5 DD 75 00                LD      (IX),L
48D8 DD 74 01                LD      (IX+1),H
48DB 3A 47 46                LD      A,(#FSTCH+2)
48DE DD 77 02                LD      (IX+2),A
48E1 E1              REVIV1: POP     HL
48E2 C3 64 46                JP      HOT
48E5
48E5                 ; Go to Line
48E5                 ;
48E5 11 FF 48        GOLINE: LD      DE,GOTOMES
48E8 CD A9 4F                CALL    GETCMND
48EB CD 78 48                CALL    BREAK?
48EE C8                      RET     Z
48EF CD 85 4F                CALL    GETLN
48F2 CD 47 4F                CALL    GETLNADR
48F5 ED 53 3F 46             LD      (#CURLN),DE
48F9 22 3D 46                LD      (#CURADR),HL
48FC C3 0C 49                JP      CURPG
48FF                 ;
48FF 47 6F 74 6F     GOTOMES:DEFB    "Goto:", 0
4903 3A 00
4905
4905                 ;
4905                 NUMCHK: ; Check if A is number
4905                 ;
4905 FE 30                   CP      '0'
4907 D8                      RET     C
4908 FE 3A                   CP      '9'+1
490A 3F                      CCF
490B C9                      RET
490C
490C                 ; Print Current Page
490C                 ;
490C                 CURPG:
490C ED 5B 3D 46             LD      DE,(#CURADR)
4910 2A 3F 46                LD      HL,(#CURLN)
4913 18 11                   JR      PRNTPG
4915
4915                 ; Print Next Page
4915                 ;
4915                 NEXTPG:
4915 ED 5B 41 46             LD      DE,(#NXTADR)
4919 1A                      LD      A,(DE)
491A B7                      OR      A
491B C8                      RET     Z
```

▶戦艦ユニットのない大戦略なんて，クリープのないコーヒーのようなものだ。だから，システムソフトさん，SUPER大戦略68Kにはぜひ戦艦ユニットを!!
阿部　貴秀（19）石川県

```
491C ED 53 3D 46            LD      (#CURADR),DE
4920 2A 43 46               LD      HL,(#NXTLN)
4923 22 3F 46               LD      (#CURLN),HL
4926
4926         PRNTPG:
4926 3E 0C                  LD      A,0CH               ; CLS
4928 CD F4 1F               CALL    PRINT
492B 3A 3C 46               LD      A,(#MAXLN)
492E 3D                     DEC     A
492F F5      PRPG1:         PUSH    AF
4930 CD 0D 4F               CALL    SETLNNO
4933 CD CD 4E               CALL    SET1LN
4936 30 13                  JR      NC,PRPG2
4938 F1                     POP     AF
4939 91                     SUB     C
493A 38 11                  JR      C,PRPG3
493C ED 53 41 46            LD      (#NXTADR),DE
4940 D5                     PUSH    DE
4941 11 00 2E               LD      DE,BUFFER
4944 CD E5 1F               CALL    MSX
4947 D1                     POP     DE
4948 23                     INC     HL
4949 18 E4                  JR      PRPG1
494B F1      PRPG2:         POP     AF
494C 1B                     DEC     DE
494D                        ;
494D 22 43 46 PRPG3:        LD      (#NXTLN),HL
4950 C9                     RET
4951         ;
4951 00 00   LSTLN: DEFW    0
4953
4953         ; Print Previous Page
4953         ;
4953         PREVPG:
4953 3A 3C 46               LD      A,(#MAXLN)
4956 3D                     DEC     A
4957 32 C0 49               LD      (SCRLN),A
495A 67                     LD      H,A                 ; save
495B ED 5B 3F 46            LD      DE,(#CURLN)
495F 1B                     DEC     DE
4960 7B                     LD      A,E
4961 B2                     OR      D
4962 C8                     RET     Z
4963 7C                     LD      A,H                 ; get
4964                        ;
4964 2A 3D 46               LD      HL,(#CURADR)
4967 22 3D 46 PRVPG1:       LD      (#CURADR),HL
496A 2B                     DEC     HL
496B 2B                     DEC     HL
496C F5                     PUSH    AF
496D CD 8C 49               CALL    PRVLN
4970 30 09                  JR      NC,PRVPG2
4972 F1                     POP     AF
4973 1B                     DEC     DE
4974 91                     SUB     C
4975 30 F0                  JR      NC,PRVPG1
4977 81                     ADD     A,C
4978 13                     INC     DE
4979 18 01                  JR      PRVPG3
497B F1      PRVPG2:        POP     AF
497C 32 C0 49 PRVPG3:       LD      (SCRLN),A
497F 13                     INC     DE
4980 ED 53 3F 46            LD      (#CURLN),DE
4984 EB                     EX      DE,HL
4985 ED 5B 3D 46            LD      DE,(#CURADR)
4989 C3 26 49 PRVPG4:       JP      PRNTPG
498C         ;
498C         PRVLN:
498C         ;
498C 7E                     LD      A,(HL)
498D B7                     OR      A
498E 20 03                  JR      NZ,PRVLN1           ; CF is clear
4990 23                     INC     HL
4991 23                     INC     HL
4992 C9                     RET
4993 3E 0D   PRVLN1:        LD      A,0DH
4995 01 00 00               LD      BC,0
4998 ED B9                  CPDR
499A 23                     INC     HL
499B 23                     INC     HL
499C E5                     PUSH    HL
499D                        ;
499D 0E 00                  LD      C,0
499F 7E      PRVLN2:        LD      A,(HL)
49A0 0C                     INC     C
49A1 23                     INC     HL
49A2 FE 0D                  CP      0DH
49A4 28 0C                  JR      Z,PRVLN3
49A6 FE 09                  CP      9                   ; TAB
49A8 20 F5                  JR      NZ,PRVLN2
49AA 79                     LD      A,C
49AB E6 F8                  AND     0F8H                ; 1111.1000
49AD C6 08                  ADD     A,8
49AF 4F                     LD      C,A
49B0 18 ED                  JR      PRVLN2
49B2                        ;
49B2 69      PRVLN3:        LD      L,C
49B3 26 00                  LD      H,0
49B5 01 04 00               LD      BC,4
49B8 09                     ADD     HL,BC
49B9 CD 04 4F               CALL    LNLEN
49BC 37                     SCF
49BD 0C                     INC     C
49BE E1                     POP     HL
49BF C9                     RET
49C0         ;
49C0 00      SCRLN:  DEFB   0
49C1
49C1         ; File I/O
49C1         ;
49C1         FILEIO:
49C1 11 C4 4A               LD      DE,IOMES
49C4 CD E5 1F               CALL    MSX
49C7 CD 21 20               CALL    FLGET
49CA E6 DF                  AND     0DFH                ; 1101,1111B
49CC FE 4C                  CP      'L'
49CE 28 14                  JR      Z,FLIO1
49D0 FE 53                  CP      'S'
49D2 28 10                  JR      Z,FLIO1
49D4 FE 43                  CP      'C'
49D6 28 0C                  JR      Z,FLIO1
49D8 FE 47                  CP      'G'
49DA 28 08                  JR      Z,FLIO1
49DC FE 44                  CP      'D'
49DE CA 01 4B               JP      Z,SELECT
49E1 C3 0C 49               JP      CURPG
49E4                        ;
49E4 F5      FLIO1:         PUSH    AF
49E5 11 E8 4A               LD      DE,NOMES
49E8 CD A9 4F               CALL    GETCMND
49EB CD 78 48               CALL    BREAK?
49EE 20 02                  JR      NZ,FLIO3
49F0 F1                     POP     AF
49F1 C9                     RET
49F2 D5      FLIO3:         PUSH    DE
49F3 21 ED 4A               LD      HL,FNAME
49F6 EB                     EX      DE,HL
49F7 01 13 00               LD      BC,19
49FA ED B0                  LDIR
49FC D1                     POP     DE
49FD 3E 04                  LD      A,4
49FF CD A3 1F               CALL    FILE
4A02 F1                     POP     AF
4A03 FE 53                  CP      'S'
4A05 28 62                  JR      Z,SAVE
4A07 FE 43                  CP      'C'
4A09 28 5E                  JR      Z,SAVE
4A0B         ;
4A0B         LOAD:
4A0B F5                     PUSH    AF
4A0C CD 09 20 LOAD1:        CALL    ROPEN
4A0F DA BA 4A               JP      C,PRERMES
4A12 28 08                  JR      Z,LOAD2
4A14 CD 9D 1F               CALL    FPRNT
4A17 CD EE 1F               CALL    LETNL
4A1A 18 F0                  JR      LOAD1
4A1C F1      LOAD2:         POP     AF
4A1D FE 4C                  CP      'L'
4A1F 28 15                  JR      Z,LOAD3
4A21 CD 33 4E               CALL    SCHEND
4A24 ED 5B 72 1F            LD      DE,(SIZE)
4A28 19                     ADD     HL,DE
4A29 38 78                  JR      C,MEMERR
4A2B EB                     EX      DE,HL
4A2C 2A 6A 1F               LD      HL,(MEMAX)
4A2F B7                     OR      A
4A30 ED 52                  SBC     HL,DE
4A32 30 1B                  JR      NC,LOAD4
4A34 18 6D                  JR      MEMERR
4A36 21 01 00 LOAD3:        LD      HL,1
4A39 22 3F 46               LD      (#CURLN),HL
4A3C 2A 08 30               LD      HL,(#TXTST)
4A3F 22 3D 46               LD      (#CURADR),HL
4A42 22 70 1F               LD      (DTADR),HL
4A45 CD A6 1F               CALL    RDD
4A48 F5                     PUSH    AF
4A49 38 6F                  JR      C,PRERMES
4A4B F1                     POP     AF
4A4C C3 E1 48               JP      REVIV1              ; RET-ADR
4A4F CD 33 4E LOAD4:        CALL    SCHEND
4A52 23                     INC     HL
4A53 22 70 1F               LD      (DTADR),HL
4A56 E5                     PUSH    HL
4A57 CD A6 1F               CALL    RDD
4A5A 38 5E                  JR      C,PRERMES
4A5C E1                     POP     HL
4A5D ED 5B 72 1F            LD      DE,(SIZE)
4A61 19                     ADD     HL,DE
4A62 2B                     DEC     HL
4A63 22 3A 46               LD      (#TXTEND),HL
4A66 C3 0C 49               JP      CURPG
4A69         ;
4A69         SAVE:
4A69 F5                     PUSH    AF
4A6A CD 33 4E               CALL    SCHEND
4A6D F1                     POP     AF
4A6E FE 43                  CP      'C'
4A70 28 06                  JR      Z,SAVE1
4A72 ED 5B 08 30            LD      DE,(#TXTST)
4A76 18 06                  JR      SAVE2
4A78 23      SAVE1:         INC     HL
4A79 ED 5B 3A 46            LD      DE,(#TXTEND)
4A7D EB                     EX      DE,HL
4A7E B7      SAVE2:         OR      A
4A7F ED 52                  SBC     HL,DE
4A81 23                     INC     HL
4A82 22 72 1F               LD      (SIZE),HL
4A85 21 00 00               LD      HL,0
4A88 22 70 1F               LD      (DTADR),HL
4A8B 21 00 00               LD      HL,0
4A8E 22 6E 1F               LD      (EXADR),HL
4A91 D5                     PUSH    DE
4A92 CD AF 1F               CALL    WOPEN
4A95 38 23                  JR      C,PRERMES
4A97 E1                     POP     HL
4A98 22 70 1F               LD      (DTADR),HL
4A9B CD AC 1F               CALL    WRD
4A9E F5                     PUSH    AF
4A9F 38 19                  JR      C,PRERMES
4AA1 F1                     POP     AF
4AA2 C9                     RET
4AA3         ;
4AA3         MEMERR:
4AA3 CD C4 1F               CALL    BELL
4AA6 CD F4 4F               CALL    CMNDLN
4AA9 11 AF 4A               LD      DE,MERMES
4AAC C3 E5 1F               JP      MSX
4AAF         ;
4AAF 20 4E 4F 20 MERMES: DEFB   " NO MEMORY", 0
4AB3 4D 45 4D 4F
4AB7 52 59 00
4ABA         ;
4ABA         PRERMES:
4ABA C1                     POP     BC
4ABB CD 33 20               CALL    ERROR
4ABE CD 21 20               CALL    FLGET
4AC1 C3 0C 49               JP      CURPG
4AC4         ;
4AC4 4C 6F 61 64 IOMES:  DEFB   "Load, Save, Get-cut, Cut-save, Dir ", 0
4AC8 2C 20 53 61
4ACC 76 65 2C 20
4AD0 47 65 74 2D
4AD4 63 75 74 2C
4AD8 20 43 75 74
4ADC 2D 73 61 76
4AE0 65 2C 20 44
4AE4 69 72 20 00
4AE8 46 49 4C 45 NOMES:  DEFB   "FILE:"
4AEC 3A
4AED 4E 4F 4E 41 FNAME:  DEFB   "NONAME             ", 0
4AF1 4D 45 20 20
4AF5 20 20 20 20
4AF9 20 20 20 20
4AFD 20 20 20 00
4B01
4B01         ; Select File Name From Dir
4B01         ;
4B01         SELECT:
4B01 11 58 4B               LD      DE,SELMES
4B04 CD E5 1F               CALL    MSX
4B07 CD 21 20               CALL    FLGET
4B0A FE 1B                  CP      1BH
4B0C C8                     RET     Z
4B0D FE 0D                  CP      0DH
4B0F 20 03                  JR      NZ,SEL1
4B11 CD 24 20               CALL    RDVSW
4B14 32 5D 1F SEL1:         LD      (DSK),A
4B17                        ;
4B17 CD EE 1F               CALL    LETNL
4B1A CD 06 20               CALL    DIR
4B1D 11 3B 4B               LD      DE,DIRMES
4B20 CD E5 1F               CALL    MSX
4B23 ED 5B 76 1F            LD      DE,(KBFAD)
4B27 CD D3 1F               CALL    GETL
4B2A 1A                     LD      A,(DE)
4B2B FE 1B                  CP      1BH
4B2D C8                     RET     Z
4B2E 21 05 00               LD      HL,5
4B31 19                     ADD     HL,DE
4B32 11 ED 4A               LD      DE,FNAME
4B35 01 13 00               LD      BC,19
4B38 ED B0                  LDIR
4B3A C9                     RET
4B3B         ;
4B3B 48 69 74 20 DIRMES: DEFB   "Hit CR on filename or BREAK^M", 0
```

▶２カ月あまりの教習所通いを終えて，やっと免許が取れたんだよ。北海道って，交通事故死ワースト１なんだって。まあ，教官の厳しいこと，厳しいこと。

蒔田　裕次（34）北海道

```
4B3F 43 52 20 6F
4B43 6E 20 66 69
4B47 6C 65 6E 61
4B4B 6D 65 20 6F
4B4F 72 20 42 52
4B53 45 41 4B 0D
4B57 00
4B58                  ;
4B58 0D 44 72 69      SELMES: DEFB    "^MDrive:", 0
4B5C 76 65 3A 00
4B60
4B60                  ; Convert Address to Line and Col
4B60                  ;
4B60                  ERRREP:
4B60 ED 5B 06 30              LD      DE,(#ERRAD)
4B64 2A 08 30         ADTOLN: LD      HL,(#TXTST)
4B67 2B                       DEC     HL
4B68 D9                       EXX
4B69 01 01 00                 LD      BC,1
4B6C D9                       EXX
4B6D                          ;
4B6D 01 00 00                 LD      BC,0
4B70 3E 0D                    LD      A,0DH
4B72 ED B1            ADTOLN1:CPIR
4B74 E5                       PUSH    HL
4B75 B7                       OR      A
4B76 ED 52                    SBC     HL,DE
4B78 E1                       POP     HL
4B79 28 14                    JR      Z,ADTOLN4
4B7B 30 05                    JR      NC,ADTOLN2
4B7D D9                       EXX
4B7E 03                       INC     BC
4B7F D9                       EXX
4B80 18 F0                    JR      ADTOLN1
4B82                          ;
4B82 3E 0D            ADTOLN2:LD      A,0DH
4B84 2B                       DEC     HL
4B85 2B                       DEC     HL
4B86 BE               ADTOLN3:CP      (HL)
4B87 2B                       DEC     HL
4B88 20 FC                    JR      NZ,ADTOLN3
4B8A D9                       EXX
4B8B 0B                       DEC     BC
4B8C D9                       EXX
4B8D 23                       INC     HL
4B8E 23                       INC     HL
4B8F                          ;
4B8F 22 AC 4E         ADTOLN4:LD      (LNADR),HL
4B92 D9                       EXX
4B93 ED 43 A3 4C              LD      (LNNO),BC
4B97 D9                       EXX
4B98 06 00                    LD      B,0
4B9A E5               ADTOLN5:PUSH    HL
4B9B B7                       OR      A
4B9C ED 52                    SBC     HL,DE
4B9E E1                       POP     HL
4B9F 28 0F                    JR      Z,ADTOLN7
4BA1 7E                       LD      A,(HL)
4BA2 FE 09                    CP      9               ; TAB
4BA4 20 06                    JR      NZ,ADTOLN6
4BA6 78                       LD      A,B
4BA7 E6 F8                    AND     0F8H            ; 1111.1000B
4BA9 C6 07                    ADD     A,7
4BAB 47                       LD      B,A
4BAC 04               ADTOLN6:INC     B
4BAD 23                       INC     HL
4BAE 18 EA                    JR      ADTOLN5
4BB0 C5               ADTOLN7:PUSH    BC
4BB1 CD E5 4D                 CALL    EDMDLN
4BB4 C1                       POP     BC
4BB5 78                       LD      A,B
4BB6 C6 05                    ADD     A,5
4BB8 6F                       LD      L,A
4BB9 26 00                    LD      H,0
4BBB 3A 5C 1F                 LD      A,(WIDTH)
4BBE 4F                       LD      C,A
4BBF CD 33 4F                 CALL    DIV
4BC2 32 A7 4C                 LD      (LOC),A
4BC5 3A A8 4C                 LD      A,(LOC+1)
4BC8 85                       ADD     A,L
4BC9 32 A8 4C                 LD      (LOC+1),A
4BCC C9                       RET
4BCD
4BCD                  ; Edit
4BCD                  ;  Continue edittig until 'BREAK'ed
4BCD                  ;
4BCD                  EDIT:
4BCD 21 04 00                 LD      HL,4
4BD0 22 A7 4C                 LD      (LOC),HL
4BD3                  ;
4BD3                  EDIT1:  ; EDIT entry
4BD3                  ;
4BD3 21 D3 4B                 LD      HL,EDIT1
4BD6 E5                       PUSH    HL
4BD7                          ;
4BD7 2A A7 4C                 LD      HL,(LOC)
4BDA CD 1E 20                 CALL    CSRSET
4BDD ED 5B 76 1F              LD      DE,(KBFAD)
4BE1 CD D3 1F                 CALL    GETL
4BE4 CD 18 20                 CALL    CSRRD
4BE7 2E 04                    LD      L,4
4BE9 22 A7 4C                 LD      (LOC),HL
4BEC 1A                       LD      A,(DE)
4BED FE 1B                    CP      1BH             ; 'BREAK'ed ?
4BEF 20 05                    JR      NZ,EDIT2
4BF1 CD F4 4F                 CALL    CMNDLN
4BF4 E1                       POP     HL
4BF5 C9                       RET
4BF6                          ;
4BF6 FE 30            EDIT2:  CP      '0'
4BF8 D8                       RET     C
4BF9 FE 3A                    CP      '9'+1
4BFB D0                       RET     NC
4BFC CD 85 4F                 CALL    GETLN
4BFF 22 A3 4C                 LD      (LNNO),HL
4C02 08                       EX      AF,AF'          ; save cntrol char
4C03 ED 53 A5 4C              LD      (CTRLCOL),DE
4C07 CD 47 4F                 CALL    GETLNADR        ; HL=ADR
4C0A 22 AC 4E                 LD      (LNADR),HL      ; save lnadrs
4C0D CD 46 4E                 CALL    ENTAB
4C10 08                       EX      AF,AF'
4C11 21 A9 4C                 LD      HL,EDTBL
4C14 06 0A                    LD      B,10
4C16 CD B2 46                 CALL    JPTBL
4C19 38 01                    JR      C,EDIT3
4C1B E9                       JP      (HL)
4C1C                          ;
4C1C 08               EDIT3:  EX      AF,AF'
4C1D 2A AC 4E                 LD      HL,(LNADR)
4C20 CD AE 4E                 CALL    LNSIZE          ; HL=SIZE
4C23 E5                       PUSH    HL
4C24 CD BC 4E                 CALL    NLSIZE          ; HL=SIZE
4C27 E5                       PUSH    HL
4C28 2A AC 4E                 LD      HL,(LNADR)
4C2B 7E                       LD      A,(HL)
4C2C E1                       POP     HL
4C2D B7                       OR      A
4C2E 20 05                    JR      NZ,EDIT4
4C30 23                       INC     HL
4C31 08                       EX      AF,AF'
4C32 3E 2B                    LD      A,'+'
4C34 08                       EX      AF,AF'
4C35                          ;
4C35 08               EDIT4:  EX      AF,AF'
4C36 FE 2B                    CP      '+'             ; Check CTRLCHR
4C38 20 04                    JR      NZ,EDIT5
4C3A 11 0A 00                 LD      DE,10
4C3D 19                       ADD     HL,DE
4C3E 08               EDIT5:  EX      AF,AF'          ; save CTRLCHR
4C3F D1                       POP     DE              ; LNSIZE
4C40 B7                       OR      A
4C41 ED 52                    SBC     HL,DE           ; HL=DIFFERENCE
4C43 7D                       LD      A,L
4C44 B4                       OR      H
4C45 28 35                    JR      Z,TRNS
4C47 EB                       EX      DE,HL
4C48 2A 41 46                 LD      HL,(#NXTADR)
4C4B 19                       ADD     HL,DE
4C4C 22 41 46                 LD      (#NXTADR),HL
4C4F EB                       EX      DE,HL
4C50 7C                       LD      A,H
4C51 E6 80                    AND     80H             ; 1000.0000B
4C53 20 0F                    JR      NZ,DECSIZE
4C55 CD 20 4E                 CALL    SIZECHK
4C58 D8                       RET     C
4C59                  ;
4C59                  INCSIZE:
4C59                  ;
4C59 CD 97 4E                 CALL    TRNSSIZE
4C5C ED 5B 3A 46              LD      DE,(#TXTEND)
4C60 ED B8                    LDDR
4C62 18 18            INCSZ1: JR      TRNS
4C64                  ;
4C64                  DECSIZE:
4C64                  ;
4C64 E5                       PUSH    HL              ; DIFFERENCE
4C65 EB                       EX      DE,HL
4C66 21 00 00                 LD      HL,0
4C69 B7                       OR      A
4C6A ED 52                    SBC     HL,DE
4C6C ED 5B AC 4E              LD      DE,(LNADR)
4C70 19                       ADD     HL,DE
4C71 E3                       EX      (SP),HL
4C72 CD 97 4E                 CALL    TRNSSIZE
4C75 ED 5B AC 4E              LD      DE,(LNADR)
4C79 E1                       POP     HL
4C7A ED B0                    LDIR
4C7C                  DECSZ1:
4C7C                  ;
4C7C                  TRNS:
4C7C                  ;
4C7C 2A AC 4E                 LD      HL,(LNADR)
4C7F CD 8E 4E                 CALL    EMPLN
4C82 28 09                    JR      Z,TRNS2
4C84 1A               TRNS1:  LD      A,(DE)
4C85 B7                       OR      A
4C86 28 05                    JR      Z,TRNS2
4C88 77                       LD      (HL),A
4C89 23                       INC     HL
4C8A 13                       INC     DE
4C8B 18 F7                    JR      TRNS1
4C8D 36 0D            TRNS2:  LD      (HL),0DH
4C8F 23                       INC     HL
4C90                          ;
4C90 08                       EX      AF,AF'
4C91 FE 2E                    CP      '.'
4C93 28 32                    JR      Z,EDMARK
4C95 FE 2B                    CP      '+'
4C97 C0                       RET     NZ
4C98 06 0A                    LD      B,10
4C9A 3E 0D                    LD      A,0DH
4C9C 77               TRNS3:  LD      (HL),A
4C9D 23                       INC     HL
4C9E 10 FC                    DJNZ    TRNS3
4CA0 C3 0C 49                 JP      CURPG
4CA3                  ;
4CA3 00 00            LNNO:   DEFW    0
4CA5 00 00            CTRLCOL:DEFW    0
4CA7 00 00            LOC:    DEFW    0
4CA9                  EDTBL:
4CA9 2D                       DEFB    '-'
4CAA 18 4D                    DEFW    DELETE
4CAC 2F                       DEFB    '/'
4CAD 18 4D                    DEFW    DELETE
4CAF 3E                       DEFB    '>'
4CB0 59 4D                    DEFW    EDYANK
4CB2 3C                       DEFB    '<'
4CB3 90 4D                    DEFW    EDDR
4CB5 3D                       DEFB    '='
4CB6 E5 4D                    DEFW    EDMDLN
4CB8 28                       DEFB    '('
4CB9 08 4D                    DEFW    SCRLD
4CBB 29                       DEFB    ')'
4CBC 0D 4D                    DEFW    SCRLU
4CBE 2C                       DEFB    ','
4CBF D0 4C                    DEFW    EXCHNGE
4CC1 5B                       DEFB    '['
4CC2 E2 4C                    DEFW    MCLTTB
4CC4 5D                       DEFB    ']'
4CC5 E2 4C                    DEFW    MCLTTB
4CC7
4CC7                  ; MARK LINE
4CC7                  ;
4CC7                  EDMARK:
4CC7 2A A3 4C                 LD      HL,(LNNO)
4CCA 22 CE 4C                 LD      (MLNNO),HL
4CCD C9                       RET
4CCE                  ;
4CCE 00 00            MLNNO:  DEFW    0
4CD0
4CD0                  ; Exchange Line and Mark
4CD0                  ;
4CD0                  EXCHNGE:
4CD0 2A CE 4C                 LD      HL,(MLNNO)
4CD3 ED 5B A3 4C              LD      DE,(LNNO)
4CD7 ED 53 CE 4C              LD      (MLNNO),DE
4CDB CD 47 4F                 CALL    GETLNADR
4CDE EB                       EX      DE,HL
4CDF C3 EC 4D                 JP      EDMDLN1
4CE2
4CE2                  ; Move Current Line to Top & Bottom
4CE2                  ;
4CE2                  MCLTTB:
4CE2 08                       EX      AF,AF'
4CE3 2A A3 4C                 LD      HL,(LNNO)
4CE6 22 3F 46                 LD      (#CURLN),HL
4CE9 2A AC 4E                 LD      HL,(LNADR)
4CEC 22 3D 46                 LD      (#CURADR),HL
4CEF 08                       EX      AF,AF'
4CF0 FE 5B                    CP      '['
4CF2 20 05                    JR      NZ,MCLTB
4CF4 CD 0C 49                 CALL    CURPG
4CF7 18 03                    JR      MCL1
4CF9 CD 53 49         MCLTB:  CALL    PREVPG
4CFC 3A 3C 46         MCL1:   LD      A,(#MAXLN)
4CFF CB 3F                    SRL     A
4D01 67                       LD      H,A
4D02 2E 04                    LD      L,4
4D04 22 A7 4C                 LD      (LOC),HL
4D07 C9                       RET
4D08
4D08                  ; Previous & Next Page
4D08                  ;
4D08                  SCRLD:
```

▶この２カ月くらいの間に，白髪がどどっと増えて増えて。この調子だと，１年後には金髪に染めなくてはならない。うーん，困った。　大脇　寿法（20）愛知県

```
4D08 CD 53 49          CALL    PREVPG
4D0B 18 03             JR      SCR1
4D0D CD 15 49  SCRLU:  CALL    NEXTPG
4D10 2A A7 4C  SCR1:   LD      HL,(LOC)
4D13 25                DEC     H
4D14 22 A7 4C          LD      (LOC),HL
4D17 C9                RET
4D18
4D18           ; Delete some lines
4D18           ;
4D18           DELETE:
4D18 08                EX      AF,AF'
4D19 CD 33 4E          CALL    SCHEND          ; Update ENDAD
4D1C 23                INC     HL
4D1D ED 5B AC 4E       LD      DE,(LNADR)
4D21 D5                PUSH    DE
4D22 1A        DELETE1:LD      A,(DE)
4D23 13                INC     DE
4D24 77                LD      (HL),A
4D25 23                INC     HL
4D26 FE 0D             CP      0DH
4D28 20 F8             JR      NZ,DELETE1
4D2A 36 00             LD      (HL),0
4D2C 22 3A 46          LD      (#TXTEND),HL
4D2F                   ;
4D2F 08                EX      AF,AF'
4D30 FE 2F             CP      '/'
4D32 20 07             JR      NZ,DELETE3
4D34 1A        DELETE2:LD      A,(DE)
4D35 13                INC     DE
4D36 FE 0D             CP      0DH
4D38 28 FA             JR      Z,DELETE2
4D3A 1B                DEC     DE
4D3B E1        DELETE3:POP     HL
4D3C D5        DELETE4:PUSH    DE
4D3D E5                PUSH    HL
4D3E B7                OR      A
4D3F ED 52             SBC     HL,DE
4D41 CD 97 4E          CALL    TRNSSIZE
4D44 D1                POP     DE
4D45 E1                POP     HL
4D46 7E                LD      A,(HL)
4D47 B7                OR      A
4D48 20 02             JR      NZ,DELETE5
4D4A 2B                DEC     HL
4D4B 03                INC     BC
4D4C ED B0     DELETE5:LDIR
4D4E CD 0C 49  DELETE6:CALL    CURPG
4D51 2A A7 4C          LD      HL,(LOC)
4D54 25                DEC     H
4D55 22 A7 4C          LD      (LOC),HL
4D58 C9                RET
4D59
4D59           ; Yank Delete Buffer
4D59           ;
4D59           EDYANK:
4D59 CD 33 4E          CALL    SCHEND          ; ENDAD
4D5C 23                INC     HL
4D5D ED 5B 3A 46       LD      DE,(#TXTEND)
4D61 D5                PUSH    DE
4D62 EB                EX      DE,HL
4D63 B7                OR      A
4D64 ED 52             SBC     HL,DE           ; SIZE
4D66 CD 20 4E          CALL    SIZECHK
4D69 D1                POP     DE
4D6A 30 06             JR      NC,EDYANK1
4D6C CD C4 1F          CALL    BELL
4D6F C3 A3 4A          JP      MEMERR
4D72                   ;
4D72 7D        EDYANK1:LD      A,L
4D73 B4                OR      H
4D74 C8                RET     Z
4D75                   ;
4D75 D5                PUSH    DE
4D76 CD 97 4E          CALL    TRNSSIZE
4D79 ED 5B 3A 46       LD      DE,(#TXTEND)
4D7D ED B8             LDDR
4D7F E1                POP     HL
4D80 ED 5B AC 4E       LD      DE,(LNADR)
4D84 7E        EDYANK2:LD      A,(HL)
4D85 B7                OR      A
4D86 28 05             JR      Z,EDYANK3
4D88 23                INC     HL
4D89 12                LD      (DE),A
4D8A 13                INC     DE
4D8B 18 F7             JR      EDYANK2
4D8D C3 0C 49  EDYANK3:JP      CURPG
4D90
4D90           ; Delete Region
4D90           ;
4D90 2A CE 4C  EDDR:   LD      HL,(MLNNO)
4D93 CD 47 4F          CALL    GETLNADR
4D96 EB                EX      DE,HL
4D97 2A AC 4E          LD      HL,(LNADR)
4D9A E5                PUSH    HL
4D9B D5                PUSH    DE
4D9C B7                OR      A
4D9D ED 52             SBC     HL,DE
4D9F 28 3F             JR      Z,EDDR3
4DA1 30 0D             JR      NC,EDDR1
4DA3                   ;
4DA3 2A A3 4C          LD      HL,(LNNO)       ; case of MARK > LN
4DA6 22 CE 4C          LD      (MLNNO),HL
4DA9 E1                POP     HL
4DAA D1                POP     DE
4DAB E5                PUSH    HL
4DAC D5                PUSH    DE
4DAD B7                OR      A
4DAE ED 52             SBC     HL,DE
4DB0                   ;
4DB0 4D        EDDR1:  LD      C,L
4DB1 44                LD      B,H             ; SIZE
4DB2 CD 20 4E          CALL    SIZECHK
4DB5 30 03             JR      NC,EDDR2
4DB7 E1                POP     HL
4DB8 E1                POP     HL
4DB9 C9                RET
4DBA CD 33 4E  EDDR2:  CALL    SCHEND
4DBD 23                INC     HL
4DBE EB                EX      DE,HL           ; DE=BUF
4DBF ED B0             LDIR
4DC1 AF                XOR     A
4DC2 12                LD      (DE),A
4DC3 ED 53 3A 46       LD      (#TXTEND),DE
4DC7 E1                POP     HL
4DC8 D1                POP     DE
4DC9 3E C9             LD      A,0C9H          ; RET
4DCB 32 4E 4D          LD      (DELETE6),A
4DCE CD 3C 4D          CALL    DELETE4
4DD1 3E CD             LD      A,0CDH          ; CALL
4DD3 32 4E 4D          LD      (DELETE6),A
4DD6                   ;
4DD6 2A CE 4C          LD      HL,(MLNNO)
4DD9 CD 47 4F          CALL    GETLNADR
4DDC EB                EX      DE,HL
4DDD C3 EC 4D          JP      EDMDLN1
4DE0           ;
4DE0 D1        EDDR3:  POP     DE
4DE1 E1                POP     HL
4DE2 C3 C4 1F          JP      BELL
4DE5
4DE5           ; Print line at middle
4DE5           ;
4DE5           EDMDLN:
4DE5 2A A3 4C          LD      HL,(LNNO)
4DE8 ED 5B AC 4E       LD      DE,(LNADR)
4DEC 22 3F 46  EDMDLN1:LD      (#CURLN),HL
4DEF ED 53 3D 46       LD      (#CURADR),DE
4DF3 3A 3C 46          LD      A,(#MAXLN)
4DF6 F5                PUSH    AF
4DF7 CB 3F             SRL     A
4DF9 F5                PUSH    AF
4DFA 32 3C 46          LD      (#MAXLN),A
4DFD 3E C9             LD      A,0C9H          ; RET
4DFF 32 89 49          LD      (PRVPG4),A
4E02 CD 53 49          CALL    PREVPG
4E05 3E C3             LD      A,0C3H          ; JP
4E07 32 89 49          LD      (PRVPG4),A
4E0A F1                POP     AF
4E0B 47                LD      B,A
4E0C 3A C0 49          LD      A,(SCRLN)
4E0F ED 44             NEG
4E11 80                ADD     A,B
4E12 3D                DEC     A
4E13 67                LD      H,A
4E14 2E 04             LD      L,4             ; (4,H)
4E16 22 A7 4C          LD      (LOC),HL
4E19 F1                POP     AF
4E1A 32 3C 46          LD      (#MAXLN),A
4E1D C3 0C 49          JP      CURPG
4E20
4E20           ; Size Check
4E20           ;   in:HL<-SIZE
4E20           ;
4E20           SIZECHK:
4E20 E5                PUSH    HL
4E21 D5                PUSH    DE
4E22 ED 5B 3A 46       LD      DE,(#TXTEND)
4E26 19                ADD     HL,DE
4E27 38 07             JR      C,SZCHK1
4E29 EB                EX      DE,HL
4E2A 2A 6A 1F          LD      HL,(MEMAX)
4E2D B7                OR      A
4E2E ED 52             SBC     HL,DE
4E30 D1        SZCHK1: POP     DE
4E31 E1                POP     HL
4E32 C9                RET
4E33
4E33           ; Search Source End
4E33           ;
4E33           SCHEND:
4E33 C5                PUSH    BC
4E34 2A 3A 46          LD      HL,(#TXTEND)
4E37 2B                DEC     HL
4E38 01 00 00          LD      BC,0
4E3B AF                XOR     A
4E3C ED B9             CPDR
4E3E 22 44 4E          LD      (ENDAD),HL
4E41 23                INC     HL
4E42 C1                POP     BC
4E43 C9                RET
4E44           ;
4E44 00 00     ENDAD:  DEFW    0
4E46
4E46           ; Entab
4E46           ;
4E46           ENTAB:
4E46 CD 8E 4E          CALL    EMPLN
4E49 C8                RET     Z
4E4A                   ;
4E4A 08                EX      AF,AF'
4E4B B7                OR      A
4E4C 08                EX      AF,AF'          ; AF' = FLAG
4E4D 0E 00             LD      C,0             ; COUNTER
4E4F 6B                LD      L,E
4E50 62                LD      H,D
4E51 1A        ENTAB1: LD      A,(DE)
4E52 77                LD      (HL),A
4E53 B7                OR      A
4E54 C8                RET     Z
4E55 FE 22             CP      '"'
4E57 28 04             JR      Z,ENTAB2
4E59 FE 27             CP      "'"
4E5B 20 03             JR      NZ,ENTAB3
4E5D 08        ENTAB2: EX      AF,AF'
4E5E 37                SCF
4E5F 08                EX      AF,AF'
4E60 FE 20     ENTAB3: CP      ' '
4E62 20 04             JR      NZ,ENTAB4
4E64 CD 6D 4E          CALL    PACK
4E67 08                EX      AF,AF'
4E68 23        ENTAB4: INC     HL
4E69 13                INC     DE
4E6A 0C                INC     C
4E6B 18 E4             JR      ENTAB1
4E6D           ;
4E6D           PACK:
4E6D           ;
4E6D 08                EX      AF,AF'
4E6E D8                RET     C               ; FLAG CHECK
4E6F 08                EX      AF,AF'
4E70                   ;
4E70 E5                PUSH    HL
4E71 79                LD      A,C
4E72 E6 07             AND     7               ; 0000.0111B
4E74 ED 44             NEG
4E76 C6 08             ADD     A,8
4E78 47                LD      B,A
4E79 1A        PACK1:  LD      A,(DE)
4E7A 77                LD      (HL),A
4E7B FE 20             CP      ' '
4E7D 20 0C             JR      NZ,PACK2
4E7F 13                INC     DE
4E80 23                INC     HL
4E81 0C                INC     C
4E82 10 F5             DJNZ    PACK1
4E84 1B                DEC     DE
4E85 0D                DEC     C
4E86 E1                POP     HL
4E87 36 09             LD      (HL),9          ; TAB
4E89 08                EX      AF,AF'
4E8A C9                RET
4E8B F1        PACK2:  POP     AF
4E8C 08                EX      AF,AF'
4E8D C9                RET
4E8E
4E8E           ; Check line is empty
4E8E           ;
4E8E           EMPLN:
4E8E ED 5B A5 4C       LD      DE,(CTRLCOL)
4E92 1B                DEC     DE
4E93 1A                LD      A,(DE)
4E94 13                INC     DE
4E95 B7                OR      A
4E96 C9                RET
4E97
4E97           ; Get transfer size
4E97           ;   BC=SIZE HL=TXTEND
4E97           ;
4E97           TRNSSIZE:
4E97 ED 5B 3A 46       LD      DE,(#TXTEND)
4E9B 19                ADD     HL,DE           ; HL=NEWEND
4E9C 22 3A 46          LD      (#TXTEND),HL
4E9F 2A AC 4E          LD      HL,(LNADR)
```

▶毎月，発売を楽しみにしております。難解な点がかなりあり，もう少し初心者にもわかりやすい内容をお願いします。ですからお願いします「おじさん特集」を!! だめでしょうか。

一関　正明 (29) 東京都

```
4EA2 D5                PUSH   DE             ; save TXTEND
4EA3 EB                EX     DE,HL
4EA4 B7                OR     A
4EA5 ED 52             SBC    HL,DE
4EA7 23                INC    HL
4EA8 4D                LD     C,L
4EA9 44                LD     B,H            ; BC=TrnsSize
4EAA E1                POP    HL
4EAB C9                RET
4EAC          ;
4EAC 00 00    LNADR:   DEFW   0
4EAE
4EAE          ; Get line length, pointed by HL, to HL
4EAE          ;
4EAE          LNSIZE:
4EAE E5                PUSH   HL
4EAF 7E       LNSIZE1:LD      A,(HL)
4EB0 23                INC    HL
4EB1 B7                OR     A
4EB2 28 04             JR     Z,LNSIZE2
4EB4 FE 0D             CP     0DH
4EB6 20 F7             JR     NZ,LNSIZE1
4EB8                   ;
4EB8 D1       LNSIZE2:POP     DE
4EB9 ED 52             SBC    HL,DE
4EBB C9                RET
4EBC
4EBC          ; Get new line length
4EBC          ;
4EBC          NLSIZE:
4EBC CD 8E 4E          CALL   EMPLN
4EBF 6B                LD     L,E
4EC0 62                LD     H,D
4EC1 23                INC    HL
4EC2 28 06             JR     Z,NLSIZE2
4EC4 2B                DEC    HL
4EC5 7E       NLSIZE1:LD      A,(HL)
4EC6 23                INC    HL
4EC7 B7                OR     A
4EC8 20 FB             JR     NZ,NLSIZE1
4ECA                   ;
4ECA ED 52    NLSIZE2:SBC     HL,DE
4ECC C9                RET
4ECD
4ECD          ; Set 1 line to buffer
4ECD          ;
4ECD          SET1LN:
4ECD E5                PUSH   HL
4ECE 21 05 2E          LD     HL,BUFFER+5
4ED1 0E 00             LD     C,0            ; COLUM
4ED3 71                LD     (HL),C
4ED4 1A       ST1LN1: LD      A,(DE)
4ED5 B7                OR     A
4ED6 28 29             JR     Z,ST1LN6
4ED8 13                INC    DE
4ED9 FE 09             CP     9              ; TAB
4EDB 20 12             JR     NZ,ST1LN3
4EDD 79                LD     A,C
4EDE E6 07             AND    7              ; 0000.0111B
4EE0 ED 44             NEG
4EE2 C6 08             ADD    A,8            ; A=8-A
4EE4 3D                DEC    A
4EE5 47                LD     B,A
4EE6 3E 20             LD     A,' '
4EE8 28 05             JR     Z,ST1LN3
4EEA 77       ST1LN2: LD      (HL),A
4EEB 0C                INC    C
4EEC 23                INC    HL
4EED 10 FB             DJNZ   ST1LN2
4EEF 77       ST1LN3: LD      (HL),A
4EF0 23                INC    HL
4EF1 0C                INC    C
4EF2 FE 0D             CP     0DH            ; CR
4EF4 20 DE             JR     NZ,ST1LN1
4EF6 36 00             LD     (HL),0
4EF8                   ;
4EF8 01 01 2E          LD     BC,BUFFER+1
4EFB ED 42             SBC    HL,BC
4EFD CD 04 4F          CALL   LNLEN
4F00 37                SCF
4F01 0C       ST1LN6: INC     C
4F02 E1                POP    HL
4F03 C9                RET
4F04          ;
4F04          LNLEN:
4F04          ;
4F04 3A 5C 1F          LD     A,(WIDTH)
4F07 4F                LD     C,A
4F08 CD 33 4F          CALL   DIV
4F0B 4D                LD     C,L
4F0C C9                RET
4F0D
4F0D          ; Set LineNo.
4F0D          ;
4F0D          SETLNNO:
4F0D F5                PUSH   AF
4F0E C5                PUSH   BC
4F0F D5                PUSH   DE
4F10 E5                PUSH   HL
4F11                   ;
4F11 06 04             LD     B,4            ; loop counter
4F13 11 04 2E          LD     DE,BUFFER+4
4F16 3E 20             LD     A,' '
4F18 12                LD     (DE),A
4F19 1B                DEC    DE
4F1A E5                PUSH   HL
4F1B CD 2E 4F PLNO1:   CALL   DIV10
4F1E C6 30             ADD    A,'0'
4F20 12                LD     (DE),A
4F21 1B                DEC    DE
4F22 10 F7             DJNZ   PLNO1
4F24 13                INC    DE
4F25 E1                POP    HL
4F26 23                INC    HL
4F27                   ;
4F27 E1                POP    HL
4F28 D1                POP    DE
4F29 C1                POP    BC
4F2A F1                POP    AF
4F2B C9                RET
4F2C          ;
4F2C 00 00    #LNNO:   DEFW   0
4F2E
4F2E          ; DIVIDED BY 10 or C
4F2E          ;
4F2E          DIV10:
4F2E C5                PUSH   BC
4F2F 0E 0A             LD     C,10
4F31 18 01             JR     DIV100
4F33 C5       DIV:     PUSH   BC
4F34                   ;
4F34 AF       DIV100:  XOR    A              ; A=0
4F35 06 10             LD     B,16           ; loop counter
4F37 29       DIV101:  ADD    HL,HL          ; shift left HL
4F38 8F                ADC    A,A
4F39 B9                CP     C
4F3A 30 05             JR     NC,DIV102
4F3C 10 F9             DJNZ   DIV101
4F3E C3 45 4F          JP     DIV103
4F41                   ;
4F41 91       DIV102:  SUB    C              ; A-10 or C
4F42 23                INC    HL             ; SetAnsBit
4F43 10 F2             DJNZ   DIV101
4F45                   ;
4F45 C1       DIV103:  POP    BC
4F46 C9                RET
4F47
4F47          ; Get ADRS of HL Line
4F47          ;
4F47          GETLNADR:
4F47 7D                LD     A,L
4F48 B4                OR     H
4F49 20 03             JR     NZ,GETLAD1
4F4B 21 01 00          LD     HL,1
4F4E ED 5B 3F 46 GETLAD1:LD   DE,(#CURLN)
4F52 2B                DEC    HL
4F53 EB                EX     DE,HL
4F54 B7                OR     A
4F55 ED 52             SBC    HL,DE          ; CURLN > GETLN ?
4F57 13                INC    DE
4F58 D5                PUSH   DE
4F59 D9                EXX
4F5A D1                POP    DE             ; save target line
4F5B D9                EXX
4F5C 38 08             JR     C,GETLAD2
4F5E 11 01 00          LD     DE,1
4F61 2A 08 30          LD     HL,(#TXTST)
4F64 18 07             JR     GETLAD3
4F66 ED 5B 3F 46 GETLAD2:LD   DE,(#CURLN)
4F6A 2A 3D 46          LD     HL,(#CURADR)
4F6D                   ;
4F6D 7E       GETLAD3:LD      A,(HL)
4F6E B7                OR     A
4F6F C8                RET    Z
4F70 CD 7D 4F          CALL   CPLN
4F73 C8                RET    Z
4F74 7E       GETLAD4:LD      A,(HL)
4F75 23                INC    HL
4F76 FE 0D             CP     0DH
4F78 20 FA             JR     NZ,GETLAD4
4F7A 13                INC    DE
4F7B 18 F0             JR     GETLAD3
4F7D          ;
4F7D          CPLN:    ; Check Curln is Targetln
4F7D          ;
4F7D D5                PUSH   DE
4F7E D9                EXX
4F7F E1                POP    HL
4F80 B7                OR     A
4F81 ED 52             SBC    HL,DE
4F83 D9                EXX
4F84 C9                RET
4F85
4F85          ; Get Line No.
4F85          ;
4F85          GETLN:
4F85 C5                PUSH   BC
4F86 21 00 00          LD     HL,0           ; ANS
4F89 3E 04             LD     A,4            ; LOOP COUNTER
4F8B F5       GETLN1:  PUSH   AF
4F8C 1A                LD     A,(DE)
4F8D 13                INC    DE
4F8E CD 05 49          CALL   NUMCHK
4F91 38 13             JR     C,GETLN2
4F93 29                ADD    HL,HL
4F94 4D                LD     C,L
4F95 44                LD     B,H
4F96 29                ADD    HL,HL
4F97 29                ADD    HL,HL
4F98 09                ADD    HL,BC
4F99 D6 30             SUB    '0'
4F9B 4F                LD     C,A
4F9C 06 00             LD     B,0
4F9E 09                ADD    HL,BC
4F9F F1                POP    AF
4FA0 3D                DEC    A
4FA1 20 E8             JR     NZ,GETLN1
4FA3 1A                LD     A,(DE)
4FA4 13                INC    DE
4FA5 F5                PUSH   AF
4FA6 C1       GETLN2:  POP    BC
4FA7 C1                POP    BC
4FA8 C9                RET
4FA9
4FA9          ; Get Command
4FA9          ;
4FA9          GETCMND:
4FA9 CD F4 4F          CALL   CMNDLN
4FAC 3E 20             LD     A,' '
4FAE 06 23             LD     B,35
4FB0 CD F4 1F CLRLN:   CALL   PRINT
4FB3 10 FB             DJNZ   CLRLN
4FB5                   ;
4FB5 CD F4 4F          CALL   CMNDLN
4FB8 CD D3 4F          CALL   PRMES
4FBB 1A       GETCM1:  LD     A,(DE)
4FBC 13                INC    DE
4FBD 2C                INC    L
4FBE FE 3A             CP     ':'
4FC0 20 F9             JR     NZ,GETCM1
4FC2 E5                PUSH   HL
4FC3 CD 1E 20          CALL   CSRSET
4FC6 ED 5B 76 1F       LD     DE,(KBFAD)
4FCA CD D3 1F          CALL   GETL
4FCD E1                POP    HL
4FCE 26 00             LD     H,0
4FD0 19                ADD    HL,DE
4FD1 EB                EX     DE,HL
4FD2 C9                RET
4FD3          ;
4FD3          PRMES:
4FD3          ;
4FD3 F5                PUSH   AF
4FD4 D5                PUSH   DE
4FD5 1A       PRMES1:  LD     A,(DE)
4FD6 13                INC    DE
4FD7 B7                OR     A
4FD8 28 17             JR     Z,PRMES3
4FDA FE 5E             CP     '^'
4FDC CC F4 1F          CALL   Z,PRINT
4FDF FE 20             CP     20H
4FE1 30 09             JR     NC,PRMES2
4FE3 F5                PUSH   AF
4FE4 3E 5E             LD     A,'^'
4FE6 CD F4 1F          CALL   PRINT
4FE9 F1                POP    AF
4FEA C6 40             ADD    A,'@'
4FEC CD F4 1F PRMES2:  CALL   PRINT
4FEF 18 E4             JR     PRMES1
4FF1 D1       PRMES3:  POP    DE
4FF2 F1                POP    AF
4FF3 C9                RET
4FF4          ;
4FF4          CMNDLN:
4FF4          ;
4FF4 3A 5B 1F          LD     A,(MAXLN)
4FF7 3D                DEC    A
4FF8 2E 00             LD     L,0            ; COL
4FFA 67                LD     H,A            ; LINE
4FFB C3 1E 20          JP     CSRSET
```

特別付録

X1版S-OS“SWORD”〈再掲載〉

編集室

数カ月前から“SWORD”システム掲載号が入手不能となっていましたが，その後の新規ユーザーに対応するため X1 版に限り特別再掲載を行うことになりました。今月掲載のアセンブラを始め，S-OS はマシンを活用するうえで欠かせないシステムです。ぜひとも入力するようにしてください。

入力方法

BASIC CZ-8FB01ver. 1.0を立ち上げ，

CLEAR &HB000

を実行したあと，MACINTO-C (B000H) を起動するか，

MON

でマシン語モニタに入りMコマンドでリスト1を入力してください。

SAVE “SWORD”, &HC500,&HDCD4

でセーブしたあと，

CALL &HC4F0

とするとS-OS “SWORD” が起動します。電源を入れるたびにBASICから立ち上げるのは面倒なので，まず“SWORD”のシステムディスクを作成しましょう。

“SWORD”が起動したままの状態でリスト2を打ち込みます。X1の場合， BASICと“SWORD”のディスクが共用できますので，X1用のMACINTO-Cをそのまま使うこともできます。

BASICのユーティリティでフォーマットを行った2Dのディスクを用意し，MACINTO-C の入ったディスクをドライブ0に入れてください。

#DA：

でディスクドライブ0のディレクトリがとれますので，

Bin A：MACINTO-C ～

の部分を，

#L A：MACINTO-C ～

のように打ち換えてリターンキーを押します。これで MACINTO-C がメモリ上にロードされました。MACINTO-C を持っていない人は，

#M

でモニタに入りMコマンドで打ち込んでください。リスト2を打ち込み終わったら，

#J7000

でシステムジェネレータを起動します。メニューの2番を選択してシステムジェネレートを行ってください（デバイス名はA, B, Cのようにアルファベットで指定します)。これでIPL起動の“SWORD”ディスクができあがりました。

ただし，このままではBASICがなくなっているため，モニタのRコマンドでの戻り先がありませんので，システムジェネレータを起動する前に，

012BH FA 1F

1053H FA 1F

のようにモニタの一部を書き換えたほうがよいかもしれません。

“SWORD”の使い方

“SWORD”が起動した状態では表1のようなコマンドが使用できます。これらのコマンドは，

#DVB：

#L REDA

#J3000

のように使用します。デバイス名は，

T カセットテープ(2700bps)

S カセットテープ(2400bps)

A～D ディスクドライブ

E～F RAMディスク用（リザーブ）

となっています。

そのほか，S-OS には表3に示すようにファイル入出力など，さまざまなルーチンがあります。“SWORD”自体はマシン語モニタにすぎませんが，これまで築きあげてきた多くのソフトウェアがこの上で走ります。今回はX1版しか掲載できませんでしたが，そのほかの機種でどうしてもバックナンバーが入手できない方には掲載号のコピーサービスを実施します（“SWORD”本体のみ)。「Oh!X “SWORD” コピー」係まで170円切手を貼った返信用封筒(210×100mm以上のもの）同封のうえ申し込んでください(機種名を必ず明記すること)。

表1 S-OS“SWORD”モニタコマンド

（[ ]は省略可能であることを示す）

#D [〈デバイス名〉：]
〈デバイス名〉で指定されたデバイスのディレクトリを表示する。省略時はデフォルトのディレクトリ。

#DV 〈デバイス名〉：
デフォルトデバイスを変更する。

#J〈アドレス〉
アドレスから始まるプログラムをコールする。サブルーチン中のRETでS-OSのモニタにリターンできる。

#K 〈ファイル名〉
〈ファイル名〉で与えられたファイルを消去する。

#L 〈ファイル名〉[：〈ロードアドレス〉]
〈ファイル名〉で与えられたファイルを〈ロードアドレス〉へロードする。ロードアドレスが省略されたときには，セーブしたときのアドレスへロードする。

#M
各機種のマシン語モニタのホットスタートへジャンプする。

#N 〈ファイル名1〉：〈ファイル名2〉
〈ファイル名1〉を〈ファイル名2〉に変更する。なお，〈ファイル名2〉のデバイス指定は不要。

#S〈ファイル名〉：〈開始番地〉：〈終了番地〉[：〈実行番地〉]
〈開始番地〉から〈終了番地〉までを〈ファイル名〉でセーブする。

#ST 〈ファイル名〉：P または ：R
〈ファイル名〉で指定されたファイルにライトプロテクトをかける。その後は同一ファイルのセーブ，消去ができなくなる。プロテクトをはずすにはRを指定。

#W
画面の40字，80字モードを切り換える。

#！
ブートコマンド。

表2 S-OSのキャラクタコード

下位＼上位	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	nul		SP	0	@	P		p					タ	ミ		
1			!	1	A	Q	a	q			。	ア	チ	ム		
2			”	2	B	R	b	r			「	イ	ツ	メ		
3			#	3	C	S	c	s			」	ウ	テ	モ		
4			$	4	D	T	d	t				エ	ト	ヤ		
5			%	5	E	U	e	u				オ	ナ	ユ		
6			&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7			'	7	G	W	g	w			ァ	キ	ヌ	ラ		
8			(	8	H	X	h	x			ィ	ク	ネ	リ		
9			)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A			*	:	J	Z	j	z			ェ	コ	ハ	レ		
B		BRK	+	;	K	[	k	■			ォ	サ	ヒ	ロ		
C	CLS	→	,	<	L	＼	l				ャ	シ	フ	ワ		
D	CR	←	−	=	M	]	m				ュ	ス	ヘ	ン		
E		↑	.	>	N	^	n				ョ	セ	ホ	゛		
F		↓	/	?	O		o	π			ッ	ソ	マ	゜		

表3　S-OSのサブルーチン一覧表

ルーチン名（アドレス）	サブルーチンの機能	レジスタ破壊
#COLD (1FFDH)	S-OSのコールドスタート。初期設定後メッセージを出力し，ワークエリア#USRに格納されているアドレスにジャンプする。#USRには初期値として#HOTアドレスが格納されている。	—
#HOT (1FFAH)	S-OSのモニタになっており，プロンプト#が出てコマンド入力待ちになる。	—
#VER (1FF7H)	HLレジスタにS-OSの機種とバージョンを返す。Hレジスタは機種を表しており，上位4ビットで機種の系列を示し，下位4ビットで系列内の機種番号を示す。 0　0　MZ-80K/C/1200　　4　0　FM-7/77 0　1　MZ-700　　5　0　SMC-777 0　2　MZ-1500　　6　0　PASOPIA 1　0　MZ-80B　　6　1　PASOPIA7 1　1　MZ-2000/2200 1　2　MZ-2500 2　0　X1 2　1　X1turbo 3　0　PC-8801(ROM) 3　1　PC-8001 3　2　PC-8801(オールRAM) LレジスタはS-OSのバージョンNo.を示しており，各種パッケージを追加したりした場合のS-OSのバージョンをチェックできるようにする。	HL
#PRINT (1FF4H)	Aレジスタの内容をアスキーコードとみなし表示する（1文字表示）。	F
#PRNTS (1FF1H)	スペースをひとつ表示する。	F
#LTNL (1FEEH)	改行する。	なし
#NL (1FEBH)	カーソルが行の先頭になければ改行する。	なし
#MSG (1FE8H)	DEレジスタの示すアドレスから0DHがあるまでASCIIコードとみなし文字列表示する。	F
#MSX (1FE5H)	DEレジスタの示すアドレスから00Hがあるまでのお ASCIIコードとみなし文字列表示する。	F
#MPRNT (1FE2H)	これをコールした次のアドレスから00HがあるまでASCIIコードとみなし文字列表示する。 例）　CALL　#MPRNT 　　　DM　"MESSAGE" 　　　DB　0	AF DE
#TAB (1FDFH)	Bレジスタの値とカーソルX座標との差だけスペースを表示する。	AF
#LPRNT (1FDCH)	Aレジスタの内容をASCIIコードとみなしプリンタのみに出力する。プリンタエラーがあった場合は，キャリフラグをセットしてリターンする。	AF
#LPTON (1FD9H)	上記#PRINT～#TAB，#PRTHX，#PRTHLの出力をディスプレイだけでなくプリンタにも出力するかどうかのフラグ#LPTSWをセットする。これをコールしたあとは，上記サブルーチンでプリンタにも出力される。	なし
#LPTOF (1FD6H)	フラグ#LPTSWをリセットする。これをコールしたあとは，#PRINT～#TAB，#PRTHX，#PRTHLの出力をディスプレイのみにする。	なし
#GETL (1FD3H)	DEレジスタにキー入力バッファの先頭アドレスを入れてコールすると，キーボードから1行入力をして文字列をバッファに格納しリターンする。エンドコードは00H。途中でSHIFT＋BREAKが押されたら，バッファ先頭に1BHが格納される。	AF
#GETKY (1FD0H)	キーボードからリアルタイムキー入力をする。入力したデータはAレジスタに格納され，何も押されていないときはAレジスタに0をセットしてリターンする。	AF
#BRKEY (1FCDH)	ブレイクキーが押されているかどうかをチェックする。押されているときはゼロフラグをセットしてリターンする。	AF
#INKEY (1FCAH)	なにかキーを押すまでキー入力待ちをし，キー入力があるとリターンする。押されたキーのASCIIコードはAレジスタにセットされる。	AF
#PAUSE (1FC7H)	スペースが押されていれば，再び何かキーを押すまでリターンしない。このときSHIFT＋BREAKを押すと，このルーチンをコールした次のアドレスの2バイトの内容を参照し，そこへジャンプする。 例）　CALL　#PAUSE 　　　DW　BRKJOB 　　　ここでブレイクキーを押すとBRKJOBへジャンプ， 　　　さもなくばDW BRKJOBはスキップ。	AF
#BELL (1FC4H)	ベル（ビープ音）を鳴らす。	AF
#PRTHX (1FC1H)	Aレジスタの内容を16進数2桁で表示する。	AF
#PRTHL (1FBEH)	HLレジスタの内容を16進数4桁で表示する。	AF
#ASC (1FBBH)	Aレジスタの下位4ビットの値を16進数を表すASCIIコードに変換し，Aレジスタにセットする。	AF
#HEX (1FB8H)	Aレジスタの内容を16進数を表すASCIIコードとしてバイナリに変換し，Aレジスタにセットする。Aレジスタの内容が16進数を表すASCIIコードでない場合は，キャリフラグをセットしてリターンする。	AF
#2HEX (1FB5H)	DEレジスタの示すアドレスから2バイトの内容を，2桁の16進数を表すASCIIコードとしてバイナリに変換し，Aレジスタにセットする。エラーの場合はキャリフラグがセットされる。	AF DE(+2)
#HLHEX (1FB2H)	DEレジスタの示すアドレスから4バイトの内容を，4桁の16進数を表すASCIIコードとしてバイナリに変換し，HLレジスタにセットする。エラーがあった場合は，キャリフラグがセットされる。	AF HL DE(+4)
#WOPEN (1FAFH)	#FILEでセットされたファイル名，(#DTADR)，(#SIZE)，(#EXADR)をテープに書き込む。ディスクの場合は，新しいファイルかどうかのチェックを行う。エラー発生時にはキャリフラグが立つ。	AF BC DE HL
#WRD (1FACH)	(#DTADR)，(#SIZE)，(#EXADR)に従って，デバイスにデータをセーブする。ディスクの場合#WOPEN後でないとFile not Openのエラーが出る。	〃
#FCB (1FA9H)	テープの場合"MACE"の#RDIとまったく同じ。ディスクの場合#DIRNOの値に従って(#IBFAD)にディレクトリの内容を転送する。これにより"MACE"用プログラムにまったく手を加えることなくディスクリードを行うことができる。CALL後，(#DIRNO)はインクリメントされる。ブレイクキーが押されると(#DIRNO)をクリアする。リターンキーが押されるとキャリフラグを立ててリターンする。	〃
#RDD (1FA6H)	(#DTADR)，(#SIZE)，(#EXADR)に従って，デバイス上のファイルを読み込む。#ROPEN後でないとFile not Openのエラーが出る。	〃
#FILE (1FA3H)	Aレジスタにファイルのアトリビュート，DEレジスタにファイル名の入っている先頭アドレスをセットしてコールすると(#IBFAD)にファイル名のセットと(#DSK)にファイルディスクリプタのセットを行う。ファイルを操作する前には，必ずこのサブルーチンにより，ファイル名とアトリビュートをセットしなければならない。コール後DEレジスタは行の終わり(00H)か：(コロン)の位置を示している。	〃
#FSAME (1FA0H)	#FILEでセットされたファイルネームと，読み込んだファイルネームを比較する。一致すればゼロ，不一致ならばノンゼロでリターンする。アトリビュートのチェックも同時に行う。	〃
#FPRNT (1F9DH)	デバイスから読み込んだファイルネームを表示する。スペースキーを押すと表示後一時停止する。	〃
#POKE (1F9AH)	HLレジスタの内容をオフセットアドレスとして，S-OS用特殊ワークエリアにAレジスタの内容を書き込む。	なし
#POKE@ (1F97H)	メインメモリからS-OS用特殊ワークエリアにデータを転送する。HLレジスタにメモリ先頭アドレス，DEレジスタにワークエリアオフセットアドレス，BCレジスタにバイト数を入れてコールする。	AF BC DE HL
#PEEK (1F94H)	HLレジスタの内容をオフセットアドレスとして，S-OS用特殊ワークエリアからAレジスタにデータを読み出す。#POKEと逆の動作。	AF
#PEEK@ (1F91H)	S-OS用特殊ワークエリアからメインメモリにデータを転送する。HL，DE，BCレジスタにセットするパラメータは#POKE@と同じ。	AF DE BC HL

#MON (1F8EH)	各機種のモニタにジャンプする。	—
[HL] (1F81H)	HLレジスタにコールしたいアドレスを入れ， CALL [HL] と使うことにより，擬似的なレジスタ間接コールが可能。	なし
#GETPC (1F80H)	現在のプログラムカウンタの値をHLにコピーする。	HL
#DRDSB (2000H)	DEが示すレコードナンバーからAが示すレコード数だけHLが示すアドレスに読み込む。連続セクタリード。(#DSK)にデバイス(A～D)をセットしてコールする。 LD DE,(#FATPOS) LD HL,(#FATBF) LD A,1 CALL #DRDSB とすれば，FATバッファにFATを読み出すことができる。	AF AF'
#DWTSB (2003H)	HLが示すアドレスからAレコード分(A×256バイト)の内容を，DEを先頭レコードとして記録する。連続セクタライト。(#DSK)にデバイス(A～D)をセットしてコール。	AF AF'
#DIR (2006H)	(#DSK)で指定されたデバイス上の全ディレクトリを表示する。	AF BC DE HL
#ROPEN (2009H)	テープの場合は，先に#FILEでセットされたファイル名と，読み込んだIBを比較し，同一ファイルならゼロ，違えばノンゼロでリターンする。ディスクの場合は，#FILEでセットされたファイルがディスク上にあるかどうかのチェックを行う。ゼロフラグは常にリセットとなる。いずれの場合にも，エラーが発生したときにはキャリでリターンする。またファイルの情報は，(#DTADR)，(#SIZE)，(#EXADR)へ転送される。	〃
#SET (200CH)	#IBFADで示されるIBバッファの内容と一致するディスク上のファイルをライトプロテクトする。	〃
#RESET (200FH)	#IBFADで示されるIBバッファの内容と一致するファイルのプロテクトをはずす。	〃
#NAME (2012H)	#FILEで設定されたファイル名を，DEレジスタが示すメモリ上のデータに変える。リネーム。メモリ上のデータ中にデバイスディスクリプタが入っていても無視する。またDE+16以内にエンドコード(00H，':')がないときにはエラーが発生する。	〃
#KILL (2015H)	#IBFADで示されるIBバッファの内容と一致するディスク上のファイルを削除する。	〃
#CSR (2018H)	現在のカーソル位置を，HにY座標，LにX座標の順で読み出す。カーソル位置の読み出しは必ずこの方法によること。(#XYADR)は使わない。	HL
#SCRN (201BH)	HにY座標，LにX座標をセットしコールすると，画面上の同位置にあるキャラクタをAに読み出す。	AF
#LOC (201EH)	HにY座標，LにX座標を入れてコールすると，カーソル位置がそこにセットされる。カーソル位置の設定は必ずこの方法によること。	AF
#FLGET (2021H)	カーソル位置で，カーソル点滅1文字入力を行い，Aに押されたキャラクタをセット。オートリピートもかかる(MZ-80K/C/1200は不可)。画面へのエコーバックは行わない。	AF
#RDVSW (2024H)	デフォルトデバイスをAに読み出す。デフォルトデバイスを知りたいときには必ずこの方法によるものとする。	A
#SDVSW (2027H)	デフォルトにしたいデバイス名をAに入れコールすると，デフォルトデバイスがセットされる。必ずこの方法によること。(#DVSW)を直接触ることを禁止する。	AF
#INP (202AH)	共通I/Oポートから1バイトをAに読み込む。ポートはCで指定する。	AF
#OUT (202DH)	共通I/OポートへAを出力する。ポートはCで指定する。	なし
#WIDCH (2030H)	画面のモード(40字，80字)を切り換える。Aに40以下の数をセットすると40字，40より大きい数をセットしてコールすると80字となる。現在のモードは(#WIDTH)に入っている。この機能はMZ-80K/C/1200/700/1500にはない。	AF BC DE HL
#ERROR (2033H)	Aにエラー番号をセットしてコールすることによりエラーメッセージを表示する。	〃

表4 S-OSのワークエリア

ワーク名 (アドレス，バイト数)	内容
#USR (1F7EH～，2バイト)	S-OSをコールドスタートしたあとジャンプするアドレスを示している。通常はS-OSのホットスタートのアドレスになっている。
#DVSW (1F7DH，1バイト)	テープフォーマットなどを切り換えるフラグ。 0：MZフォーマット2400ボー(共通モード) 1：各機種のモニタに依存 3：QD(MZ-1500のみ) コールドスタート時は0になっている。
#LPSW (1F7CH，1バイト)	#PRINT～#TAB，#PRTHX，#PRTHLルーチンでの出力をディスプレイだけでなくプリンタにも出力するかどうかのフラグ。0以外でプリンタにも出力。コールドスタート時は0になっている。
#PRCNT (1F7AH～，2バイト)	改行してから表示した文字数を格納してあるアドレスを示している。
#XYADR (1F78H～，2バイト)	カーソル座標が格納されているアドレスを示している。
#KBFAD (1F76H～，2バイト)	各機種のキー入力用バッファのアドレスを示している。 例) LD DE,(#KBFAD) CALL #GETL
#IBFAD (1F74H～，2バイト)	インフォメーションブロックの先頭アドレスを示している。同時にファイルアトリビュートのアドレスでもある。
#SIZE (1F72H～，2バイト)	ファイルサイズ。#WOPEN，#WRD，#FCB，#RDD，#ROPENルーチンで使用される。
#DTADR (1F70H～，2バイト)	ファイル先頭アドレス。
#EXADR (1F6EH～，2バイト)	ファイルのエントリアドレス。
#STKAD (1F6CH～，2バイト)	各機種のモニタが使用しているスタックのアドレスを示している。
#MEMAX (1F6AH～，2バイト)	S-OSで使用できるメモリの上限を表す。
#WKSIZ (1F68H～，2バイト)	S-OS用特殊ワークエリアのサイズを表す。
#DIRNO (1F67H，1バイト)	#FCBで使用するワーク。このワークに値を入れて#FCBをコールすると，先頭から数えてその値で示されるFCBを(#IBFAD)にロードする。ロード後，値は1増える。
#MXTRK (1F66H，1バイト)	使用できる最大トラック数が入っている。
#DTBUF (1F64H～，2バイト)	ディスクからデータを読み込む先頭アドレスが入っている。データバッファは256バイト。
#FATBF (1F62H～，2バイト)	ディスクからFATを読み込む先頭アドレスが入っている。FATバッファは256バイト。
#DIRPS (1F60H～，2バイト)	ディレクトリが入っているレコードナンバーの始まりを示す。S-OS"SWORD"では10H。書き換えることによってディレクトリの位置を移動できる。
#FATPOS (1F5EH～，2バイト)	ファイルアロケーションテーブル(FAT)が入っているレコードナンバーを示す。S-OS"SWORD"では0EH。書き換えることによりFATの位置を移動することができる。
#DSK (1F5DH，1バイト)	アクセスしようとするデバイス名が入る。
#WIDTH (1F5CH，1バイト)	現在のスクリーンモードが入っている。 40字の場合：28H 80字の場合：50H MZ-80K/C/1200/700/1500は横40字固定。
#MAXLN (1F5BH，1バイト)	画面に表示できる最大行数が入っている。

リスト1　X1版S-OS“SWORD”

```
C4F0 21 00 C5 11 00 15 01 00 : 0D
C4F8 18 ED B0 C3 FD 1F 00 00 : 94
---------------------------------
SUM: 39 ED 75 D4 FD 34 01 00 74FE


C500 CD F5 1B 22 2B 01 3E 50 : B9
C508 CD 4D 00 01 00 10 AF ED : C7
C510 79 04 ED 79 04 ED 79 32 : 7F
C518 90 0E 32 7C 1F 32 7D 1F : 39
C520 32 50 15 3C 32 A5 0E 32 : EA
C528 66 03 CD E2 1F 0C 3C 3C : BB
C530 3C 3C 3C 20 53 2D 4F 53 : F6
C538 20 20 53 57 4F 52 44 20 : EF
C540 3E 3E 3E 3E 3E 20 0D 00 : 63
C548 2A 7E 1F E9 21 20 20 C9 : DA
C550 00 F5 3E 20 18 0F F5 3E : AD
C558 0D 18 0A F5 3A 50 15 B7 : 7A
C560 20 F5 F1 C9 F5 CD 04 18 : AD
C568 F5 FE 0D 3A 50 15 20 02 : C1
C570 3E FF 3C 32 50 15 3A 7C : C6
C578 1F B7 28 05 F1 F5 CD BF : 75
---------------------------------
SUM: 7E 75 B2 23 78 EB 22 82 9122

C580 15 F1 F5 CD 13 00 F1 F1 : BD
C588 C9 F5 D5 1A FE 0D 28 12 : F2
C590 CD 64 15 13 18 F5 F5 D5 : 30
C598 1A B7 28 06 CD 64 15 13 : 58
C5A0 18 F6 D1 F1 C9 E3 7E 23 : 1D
C5A8 B7 20 02 E3 C9 CD 64 15 : CB
C5B0 18 F4 3A 50 15 90 3F D8 : 52
C5B8 CD 51 15 3C 20 FA C9 E5 : 37
C5C0 D5 C5 F5 FE 0D 20 02 3E : FA
C5C8 0A 5F 16 10 21 00 00 01 : B1
C5D0 01 1A ED 78 E6 08 28 12 : A8
C5D8 2B 7C B5 20 F5 15 20 F2 : 98
C5E0 AF 32 7C 1F F1 37 C1 D1 : 36
C5E8 E1 C9 0D 7B FE 0D 20 02 : 5F
C5F0 3E 0A ED 79 0E 03 3E 0E : 0B
C5F8 ED 79 3C ED 79 F1 B7 18 : C8
---------------------------------
SUM: 3F 94 88 06 3C 15 2D 1C 3981

C600 E5 F5 3E 01 32 7C 1F F1 : D7
C608 C9 F5 AF 32 7C 1F F1 C9 : F4
C610 CD 69 1C 30 09 EB 36 1B : C7
C618 23 36 00 2B EB C9 D5 1A : 27
C620 B7 20 02 D1 C9 CD F7 17 : 4E
C628 12 13 18 F3 3E 00 CD 1B : 56
C630 00 B7 C8 FE 03 20 03 3E : E1
C638 1B C9 C3 F7 17 CD 2C 16 : C4
C640 B7 28 FA C9 CD 4A 00 28 : E1
C648 0E CD 2C 16 FE 20 20 0E : 69
C650 CD 60 1C FE 1B 20 07 E3 : 6C
C658 7E 23 66 6F E3 C9 E3 23 : 28
C660 23 E3 C9 E5 D5 C5 F5 CD : 10
C668 F7 07 F1 C1 D1 E1 C9 7C : A7
C670 CD 74 16 7D F5 0F 0F 0F : F6
C678 0F CD 7D 16 F1 CD 83 16 : C6
---------------------------------
SUM: 88 DF A3 CC 18 DE 68 1F C24D

C680 C3 64 15 E6 0F F6 30 FE : 55
C688 3A D8 C6 07 C9 D6 30 D8 : 86
C690 FE 0A 38 07 FE 11 D8 D6 : 04
C698 07 FE 10 3F C9 C5 1A 13 : 0F
C6A0 CD 8D 16 38 0D 0F 0F 0F : E2
C6A8 0F 4F 1A 13 CD 8D 16 38 : 33
C6B0 01 B1 C1 C9 CD 9D 16 67 : 23
C6B8 D4 9D 16 6F C9 2A 72 1F : 7A
C6C0 22 92 14 2A 6E 1F 22 96 : 37
C6C8 14 2A 70 1F 22 94 14 21 : B8
C6D0 80 14 01 20 00 CD 4E 1C : EC
C6D8 B7 20 06 01 80 00 CD 44 : 6F
C6E0 17 CD 3B 00 CD AE 17 C3 : 74
C6E8 79 17 2A 70 1F ED 4B 72 : F3
C6F0 1F CD 4E 1C B7 CC 44 17 : 34
C6F8 CD 3E 00 CD AE 17 18 79 : 2E
---------------------------------
SUM: 9C 4D 68 79 70 03 0E 68 3057

C700 21 80 14 CD 4E 1C 01 20 : 0D
C708 00 B7 20 06 01 80 00 CD : 2B
C710 44 17 CD 41 00 CD AE 17 : FB
C718 E5 2A 92 14 22 72 1F 2A : 92
C720 94 14 22 70 1F 2A 96 14 : 2D
C728 22 6E 1F E1 18 4B 2A 70 : 8D
C730 1F ED 4B 72 1F CD 4E 1C : 1F
C738 B7 CC 44 17 CD 44 00 CD : BC
C740 AE 17 18 35 E5 21 94 0D : B9
C748 36 24 21 9D 0D 36 1B 21 : 97
C750 AF 0D 36 4C 21 B8 0D 36 : 5A
C758 43 21 C0 0D 36 34 21 F0 : AC
```

```
C760 0C 36 8A 21 FB 0C 36 A5 : CF
C768 21 04 0D 36 8A 21 3B 0D : 5B
C770 36 28 21 4E 0D 36 20 E1 : 11
C778 C9 E5 21 94 0D 36 20 21 : E7
---------------------------------
SUM: D8 63 6B 66 7C 3D 6A A3 12A0

C780 9D 0D 36 18 21 AF 0D 36 : 0B
C788 44 21 B8 0D 36 3C 21 C0 : 7D
C790 0D 36 2E 21 F0 0C 36 A5 : 69
C798 21 FB 0C 36 8A 21 04 0D : 1A
C7A0 36 A5 21 3B 0D 36 20 21 : BB
C7A8 4E 0D 36 28 E1 C9 F5 CD : 25
C7B0 EA 0D FB 00 F1 C9 E5 C5 : 56
C7B8 01 00 40 09 44 4D ED 79 : 41
C7C0 C1 E1 C9 E5 C5 01 00 40 : 56
C7C8 09 44 4D ED 78 18 F1 C5 : CD
C7D0 4B 3E 40 82 47 D1 7E 23 : 04
C7D8 ED 79 03 1B 7A B3 20 F6 : C7
C7E0 C9 00 00 C5 4B 3E 40 82 : D9
C7E8 47 D1 ED 78 03 77 23 1B : 35
C7F0 7A B3 20 F6 C9 00 00 E5 : F1
C7F8 C5 4F 06 00 21 16 18 09 : 72
---------------------------------
SUM: CF CD 26 8A 2A 95 59 7D 67BE

C800 7E C1 E1 C9 E5 C5 4F 06 : E8
C808 00 21 16 19 09 7E 18 F1 : E0
C810 00 00 00 00 00 00 00 00 : 00
C818 00 1B 00 00 00 00 00 00 : 1B
C820 00 00 0C 0D 00 00 00 00 : 19
C828 00 00 00 00 00 00 00 00 : 00
C830 00 1B 1C 1D 1E 1F 20 21 : D2
C838 22 23 24 25 26 27 28 29 : 2C
C840 2A 2B 2C 2D 2E 2F 30 31 : 6C
C848 32 33 34 35 36 37 38 39 : AC
C850 3A 3B 3C 3D 3E 3F 40 41 : EC
C858 42 43 44 45 46 47 48 49 : 2C
C860 4A 4B 4C 4D 4E 4F 50 51 : 6C
C868 52 53 54 55 56 57 58 59 : AC
C870 5A 5B 5C 5D 5E 5F 60 61 : EC
C878 62 63 64 65 66 67 68 69 : 2C
---------------------------------
SUM: D0 73 83 79 82 E1 0F A9 1A25

C880 6A 6B 6C 6D 6E 6F 70 71 : 6C
C888 72 73 74 75 76 77 78 79 : AC
C890 7A 7B 7C 7D 7E 7F 20 20 : 2B
C898 20 20 20 20 20 7B 20 20 : 5B
C8A0 20 20 20 20 20 20 20 20 : 00
C8A8 20 20 20 20 20 20 20 20 : 00
C8B0 20 20 20 20 20 20 A0 A1 : 01
C8B8 A2 A3 A4 A5 A6 A7 A8 A9 : 2C
C8C0 AA AB AC AD AE AF B0 B1 : 6C
C8C8 B2 B3 B4 B5 B6 B7 B8 B9 : AC
C8D0 BA BB BC BD BE BF C0 C1 : EC
C8D8 C2 C3 C4 C5 C6 C7 C8 C9 : 2C
C8E0 CA CB CC CD CE CF D0 D1 : 6C
C8E8 D2 D3 D4 D5 D6 D7 D8 D9 : AC
C8F0 DA DB DC DD DE DF 20 20 : 6B
C8F8 20 20 20 20 20 20 20 20 : 00
---------------------------------
SUM: E6 F1 FC 07 12 78 88 92 09AF

C900 20 20 20 20 20 20 7D 20 : 5D
C908 20 20 20 20 20 20 20 20 : 00
C910 20 20 20 20 20 20 00 00 : C0
C918 00 00 00 00 00 00 00 00 : 00
C920 00 00 0C 0D 00 00 00 00 : 19
C928 00 00 00 00 00 00 00 00 : 00
C930 00 00 1C 1D 1E 1F 20 21 : B7
C938 22 23 24 25 26 27 28 29 : 2C
C940 2A 2B 2C 2D 2E 2F 30 31 : 6C
C948 32 33 34 35 36 37 38 39 : AC
C950 3A 3B 3C 3D 3E 3F 40 41 : EC
C958 42 43 44 45 46 47 48 49 : 2C
C960 4A 4B 4C 4D 4E 4F 50 51 : 6C
C968 52 53 54 55 56 57 58 59 : AC
C970 5A 5B 5C 5D 5E 5F 60 61 : EC
C978 62 63 64 65 66 67 68 69 : 2C
---------------------------------
SUM: B2 BB EC F7 F4 FE 45 F2 968F

C980 6A 6B 6C 6D 6E 6F 70 71 : 6C
C988 72 73 74 75 76 77 78 79 : AC
C990 7A 87 7C F0 7E 7F 20 20 : AA
C998 20 20 20 20 20 20 20 20 : 00
C9A0 20 20 20 20 20 20 20 20 : 00
C9A8 20 20 20 20 20 20 20 20 : 00
C9B0 20 20 20 20 20 20 A0 A1 : 01
C9B8 A2 A3 A4 A5 A6 A7 A8 A9 : 2C
C9C0 AA AB AC AD AE AF B0 B1 : 6C
C9C8 B2 B3 B4 B5 B6 B7 B8 B9 : AC
C9D0 BA BB BC BD BE BF C0 C1 : EC
C9D8 C2 C3 C4 C5 C6 C7 C8 C9 : 2C
C9E0 CA CB CC CD CE CF D0 D1 : 6C
```

```
C9E8 D2 D3 D4 D5 D6 D7 D8 D9 : AC
C9F0 DA DB DC DD DE DF 20 20 : 6B
C9F8 20 20 20 20 20 20 20 20 : 00
---------------------------------
SUM: E6 FD FC 7A 12 1D 88 92 500D

CA00 20 20 20 20 20 20 20 20 : 00
CA08 20 20 20 20 20 20 20 20 : 00
CA10 20 20 20 20 20 20 CD 2B : B8
CA18 1A D5 21 70 1C 11 80 14 : 41
CA20 01 12 00 ED B0 D1 CD 12 : 60
CA28 1B B7 C9 21 70 1C 77 23 : E2
CA30 32 1F 29 CD 95 1A CD 15 : D8
CA38 29 D8 32 5D 1F 06 0D CD : 8F
CA40 86 1A 1A 20 03 3E 20 1B : 56
CA48 FE 2E 20 03 3E 20 1B 77 : 3F
CA50 13 23 10 EB 1A FE 2E 20 : 97
CA58 01 13 06 03 CD 86 1A 1A : A4
CA60 20 03 3E 20 1B 77 13 23 : 49
CA68 10 F2 36 20 3A 5D 1F CD : DB
CA70 18 29 C0 FE 53 C8 21 81 : BC
CA78 1C 06 11 7E FE 21 D0 3E : DE
---------------------------------
SUM: ED 97 3A D5 1E 1D 51 11 014E

CA80 0D 77 2B 10 F6 C9 D5 CD : 20
CA88 12 1B 1A D1 FE 3A C8 FE : 16
CA90 20 30 01 BF C9 CD 12 1B : D3
CA98 13 1A 1B FE 3A 28 04 CD : 79
CAA0 24 20 C9 1A 13 13 FE 61 : AC
CAA8 D8 FE 7B D0 D6 20 C9 11 : F1
CAB0 81 14 06 0D 1A FE 20 30 : 10
CAB8 03 3E 20 1B FE 2E 20 02 : CA
CAC0 3E 20 CD 64 15 13 10 EC : B3
CAC8 3E 2E CD 64 15 06 03 1A : D5
CAD0 FE 20 30 03 3E 20 1B CD : 97
CAD8 64 15 13 10 F2 CD 44 16 : B5
CAE0 E2 1A C9 E6 87 47 21 80 : 1A
CAE8 14 7E E6 87 B8 C2 0D 1B : A1
CAF0 3A 20 29 F5 3A 5D 1F 32 : 60
CAF8 20 29 CD 2B 1A F1 32 20 : 9E
---------------------------------
SUM: 00 B0 4D 18 E5 B4 AB 2D 9AA7

CB00 29 11 80 14 21 70 1C 06 : 81
CB08 10 CD 3D 1B C8 3E 08 B7 : FA
CB10 C9 13 1A FE 20 28 FA C9 : FF
CB18 3A 5D 1F FE 51 20 04 3E : 67
CB20 0B 37 C9 CD 32 1C 30 01 : 57
CB28 C9 21 70 1C 11 80 14 06 : 21
CB30 10 1A E6 07 BE 20 29 CD : EB
CB38 3D 1B 20 24 C9 13 23 7E : 19
CB40 FE 21 30 02 AF C9 7E FE : 45
CB48 2E 20 02 3E 20 4F 1A FE : 15
CB50 2E 20 02 3E 20 B9 C0 FE : 25
CB58 0D C8 23 13 10 E8 AF C9 : 7B
CB60 21 71 1C 7E FE 20 C8 FE : 10
CB68 0D C8 3E 05 CD EC 0D 3E : 1C
CB70 08 B7 C9 CD 32 1C 38 11 : EC
CB78 21 80 14 7E CD 12 29 CD : 08
---------------------------------
SUM: 1B 74 C3 9E ED B8 EF F3 136A

CB80 5B 15 3E 05 CD EC 0D 18 : 91
CB88 EA FE 03 20 03 3D 37 C9 : 4B
CB90 B7 C9 3E 01 C3 B7 1C E5 : 3A
CB98 CD AF 1B 38 10 C5 CD 4D : BE
CBA0 05 44 4D ED 78 C1 C3 90 : 0F
CBA8 1C 02 3E 20 B7 E1 C9 C5 : A2
CBB0 47 3A 07 00 3D BD 38 0A : C4
CBB8 3A 5B 1F 3D BC 38 03 78 : 60
CBC0 C1 C9 3E 0E C1 C9 2A 0E : 98
CBC8 00 C9 CD AF 1B D8 22 0E : 68
CBD0 00 C9 21 D3 00 22 FE FF : DC
CBD8 3E 1D C3 FE FF C5 06 00 : E6
CBE0 ED 78 C1 C9 C5 06 00 ED : A7
CBE8 79 C1 C9 CD 4D 00 3A 07 : 5E
CBF0 00 32 5C 1F C9 21 10 1C : C3
CBF8 11 6A 06 01 18 00 ED B0 : 37
---------------------------------
SUM: E1 B3 26 EC 99 EB 7B C5 6CE6

CC00 21 00 00 22 EE 06 22 89 : E2
CC08 06 22 97 08 21 00 15 C9 : C6
CC10 CB E0 D9 CB E0 D9 ED 78 : 6D
CC18 03 D9 ED 79 03 1D 20 F5 : 77
CC20 00 00 00 00 00 00 00 00 : 00
CC28 CD BD 16 18 0D CD EA 16 : 92
CC30 18 08 CD 00 17 18 03 CD : EC
CC38 2E 17 D0 E5 21 49 1C 3D : BD
CC40 85 6F 30 01 24 7E E1 37 : DF
CC48 C9 01 01 02 04 01 3A 5D : 69
CC50 1F FE 51 20 06 3E 02 E1 : B5
CC58 E1 37 C9 FE 54 C0 AF C9 : 6B
CC60 CD 2C 16 B7 20 FA C3 3D : E0
CC68 16 AF 32 50 15 C3 03 00 : 22
```

▶SOURCERYはいい。しかし，ただひとつの欠点は，その名前からソーサリアンを思い出すことだ。なんでturbo専用なんだよ。　高橋　賢一郎（17）長崎県

THE SENTINEL

```
CC70 00 00 00 00 00 00 00 00 : 00
CC78 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 39 37 A3 93 EE 64 DF 5A 7D9B

CC80 00 00 00 00 00 00 00 00 : 00
CC88 00 00 00 00 00 00 00 00 : 00
CC90 FE 20 30 02 3E 20 CD F7 : 72
CC98 17 B7 E1 C9 00 00 00 00 : 78
CCA0 CD 92 1B CD F7 17 B7 C0 : CC
CCA8 18 F6 00 00 00 00 00 00 : 0E
CCB0 32 5C 1F CD 4D 00 C9 32 : C2
CCB8 A6 0E 32 A7 0E C3 1B 00 : 79
---------------------------------
SUM: D2 C9 7D 0C 90 FA 68 E9 44AD

CF5B 19 50 41 0E 00 10 00 00 : C8
CF63 2E 00 2F 50 00 00 C0 00 : 6D
CF6B FF 00 00 00 00 00 00 00 : FF
CF73 00 80 14 00 FF 0E 00 50 : F1
CF7B 15 00 00 FA 1F E1 E9 00 : F8
CF83 00 00 00 00 00 00 00 00 : 00
CF8B 00 00 00 C3 00 10 C3 E3 : 79
CF93 17 C3 C3 17 C3 CF 17 C3 : 20
CF9B B6 17 C3 AF 1A C3 E3 1A : 19
CFA3 C3 16 1A C3 4F 23 C3 7C : 67
CFAB 23 C3 2D 23 C3 B3 22 C3 : 91
CFB3 B4 16 C3 9D 16 C3 8D 16 : A6
CFBB C3 83 16 C3 6F 16 C3 74 : DB
CFC3 16 C3 63 16 C3 44 16 C3 : 32
CFCB 3D 16 C3 4A 00 C3 2C 16 : 65
CFD3 C3 10 16 C3 09 16 C3 01 : 8F
---------------------------------
SUM: 9B 05 66 4A 5E 6D A0 B3 1064

CFDB 16 C3 BF 15 C3 B2 15 C3 : FA
CFE3 A5 15 C3 96 15 C3 89 15 : 89
CFEB C3 5B 15 C3 56 15 C3 51 : 75
CFF3 15 C3 64 15 C3 4C 15 C3 : 38
CFFB 00 21 C3 00 15 C3 44 25 : 25
D003 C3 5A 25 C3 19 24 C3 FA : FF
D00B 22 C3 08 25 C3 26 25 C3 : E3
D013 AC 24 C3 77 24 C3 C6 1B : D2
D01B C3 97 1B C3 CA 1B C3 A0 : 80
D023 1C C3 AD 25 C3 C9 25 C3 : 25
D02B DD 1B C3 E4 1B C3 EB 1B : 83
D033 C3 6C 28 C3 D2 1B       : 07
---------------------------------
SUM: A3 39 61 71 80 68 3B 67 D558

D100 ED 7B 6C 1F CD D6 1F 3E : F3
D108 23 CD F4 1F ED 5B 76 1F : E0
D110 CD D3 1F CD 1B 21 DC 33 : D7
D118 20 18 E5 1A FE 23 28 02 : 82
D120 B7 C9 13 1A 13 B7 C8 FE : 3D
D128 21 CA 36 20 FE 4A CA 72 : C5
D130 21 FE 4C CA E1 21 FE 4B : 80
D138 CA 38 22 FE 4E CA 71 22 : CD
D140 FE 4D CA 82 21 FE 57 CA : D7
D148 82 22 FE 53 28 08 FE 44 : 67
D150 28 12 3E 0D 37 C9 1A CD : 6C
D158 AA 22 13 FE 54 CA 43 22 : 60
D160 1B C3 92 21 1A CD AA 22 : 44
D168 13 FE 56 CA 5C 22 1B C3 : 8D
D170 85 21 CD 94 22 CD B2 1F : C7
D178 3E 0D D8 EB 21 00 21 E3 : 33
---------------------------------
SUM: 03 8E C1 71 A0 B6 E4 53 A65F

D180 EB E9 C3 8E 1F CD 94 22 : C7
D188 CD 9A 22 32 5D 1F CD 06 : 0A
D190 20 C9 CD 94 22 3E 01 CD : 78
D198 A3 1F 1A FE 3A 20 3E 13 : 85
D1A0 CD B2 1F 38 38 22 70 1F : BF
D1A8 22 6E 1F 13 CD B2 1F 38 : 98
D1B0 2C D5 ED 5B 70 1F B7 ED : 7C
D1B8 52 D1 38 21 23 22 72 1F : 52
D1C0 13 CD B2 1F 38 03 22 6E : 7C
D1C8 1F CD AF 1F D8 CD AC 1F : 2A
D1D0 D8 CD EB 1F 11 F3 2A CD : AA
D1D8 E8 1F C3 EB 1F 3E 0D 37 : 56
D1E0 C9 3E 01 CD A3 1F 1A B7 : 68
D1E8 32 22 22 28 09 13 CD B2 : 39
D1F0 1F 38 EA 22 20 22 CD 09 : 7B
D1F8 20 D8 C4 23 22 20 F7 CD : E5
---------------------------------
SUM: 14 27 0F 9B 9E D4 08 3B 0298

D200 E2 1F 4C 6F 61 64 69 6E : 58
D208 67 20 00 CD 9D 1F CD EB : C8
D210 1F 3A 22 22 B7 28 06 2A : AC
D218 20 22 22 70 1F C3 A6 1F : 7B
D220 00 00 00 F5 CD E2 1F 46 : 09
D228 6F 75 6E 64 20 20 20 00 : 16
D230 CD 9D 1F CD EB 1F F1 C9 : 1A
D238 CD 94 22 CD A3 1F D8 CD : B7
D240 15 20 C9 CD 94 22 CD A3 : F1
D248 1F 13 CD 94 22 1A FE 50 : 1D
D250 CA 0C 20 FE 52 CA 0F 20 : 3F
D258 3E 0D 37 C9 CD 94 22 1A : E8
D260 CD AA 22 CD 15 29 30 03 : D7
D268 3E 03 C9 32 5D 1F C3 27 : A2
D270 20 CD 94 22 CD A3 1F 1A : 4C
D278 13 FE 3A CA 12 20 3E 0D : 92
---------------------------------
SUM: 0B 05 E5 D4 75 53 36 FC DDB9

D280 37 C9 3A 5C 1F FE 50 20 : 23
D288 05 3E 28 C3 30 20 3E 50 : 0C
D290 C3 30 20 13 1A FE 20 28 : 86
D298 FA C9 CD 94 22 13 1A 1B : 8E
D2A0 FE 3A 28 03 C3 AD 25 1A : 12
D2A8 13 13 FE 61 D8 FE 7B D0 : A6
D2B0 E6 DF C9 CD 75 25 3A 5D : 8C
D2B8 1F CD 51 28 D8 CA 06 29 : 36
D2C0 CD 91 25 30 01 C9 CD FF : 49
D2C8 26 D8 CD 6B 27 20 16 7E : 11
D2D0 CD 7C 25 D8 CD 84 25 D8 : 94
D2D8 E5 01 1E 00 09 7E E1 CD : 39
D2E0 4E 27 D8 18 06 CD A2 27 : 01
D2E8 3E 09 D8 ED 53 DF 27 22 : 87
D2F0 E1 27 CD 3F 29 CD 70 25 : 9F
D2F8 AF C9 CD 75 25 3A 5D 1F : 95
---------------------------------
SUM: D0 FF 0E 4B 18 67 27 D2 3FC9

D300 CD 51 28 D8 CA 03 29 CD : E1
D308 91 25 30 01 C9 CD 6B 27 : 0F
D310 D8 3E 08 37 C0 E5 ED 5B : 42
D318 74 1F 01 20 00 ED B0 E1 : 32
D320 7E CD 84 25 D8 CD 2A 29 : EC
D328 CD 70 25 AF C9 3A 5D 1F : 90
D330 CD 51 28 D8 CA 09 29 3A : 54
D338 1E 29 B7 20 04 37 3E 0C : A3
D340 C9 CD 75 25 3A 5D 1F CD : B3
D348 91 25 D8 CD 5C 26 C9 3A : E0
D350 5D 1F CD 51 28 D8 CA 0C : 70
D358 29 AF 32 67 1F 32 18 24 : FE
D360 3A 1E 29 B7 20 04 37 3E : D1
D368 0C C9 CD 75 25 3A 5D 1F : F2
D370 CD 91 25 D8 CD FF 26 D8 : 25
D378 CD E3 25 C9 CD 75 25 3A : 3F
---------------------------------
SUM: A0 A5 75 73 7E 28 C8 64 D4BB

D380 5D 1F CD 51 28 D8 20 09 : C3
D388 CD B4 25 32 5D 1F C3 00 : 17
D390 29 CD D0 1F FE 1B CA 0D : D5
D398 24 FE 0D 20 06 3A 18 24 : CB
D3A0 B7 20 5F 3A 67 1F 4F 06 : 4B
D3A8 03 CB 3F 10 FC 2A 60 1F : C2
D3B0 16 00 5F 19 EB 2A 64 1F : 26
D3B8 3E 01 CD 44 25 38 3D 79 : 63
D3C0 E6 07 06 05 87 10 FD 2A : B6
D3C8 64 1F 85 6F 30 01 24 7E : 4A
D3D0 B7 28 13 FE FF 28 36 ED : 3A
D3D8 5B 74 1F 01 20 00 ED B0 : AC
D3E0 CD EC 23 C3 25 23 CD EC : A0
D3E8 23 30 A6 C9 21 67 1F 34 : 9D
D3F0 7E 21 66 1F BE 28 16 32 : 52
D3F8 18 24 B7 C9 F5 CD 0D 24 : AF
---------------------------------
SUM: 67 AD 3C 50 CB AF 68 B2 54DE

D400 F1 C9 21 67 1F 7E B7 28 : BE
D408 01 35 AF 18 04 AF 32 67 : 49
D410 1F 32 18 24 3E 08 37 C9 : D3
D418 00 3A 5D 1F CD 51 28 D8 : D4
D420 CA 0F 29 CD 91 25 D8 CD : 2A
D428 FF 26 D8 3E 24 CD F4 1F : 3F
D430 CD 21 27 CD C1 1F 11 99 : 6C
D438 28 CD E5 1F 06 10 ED 5B : 57
D440 60 1F 2A 64 1F 3E 01 CD : 38
D448 44 25 D8 CD 54 24 C8 13 : 61
D450 10 F0 AF C9 C5 D5 06 08 : 20
D458 7E B7 28 0F FE FF 28 12 : A3
D460 CD E3 27 CD EE 1F CD C7 : 45
D468 1F 72 24 11 20 00 19 10 : 0F
D470 E7 3E AF D1 C1 B7 C9 3A : 20
D478 5D 1F CD 9C 25 D8 CD 91 : 40
---------------------------------
SUM: 31 2A F2 0D D4 8B 85 AC B8A7

D480 25 D8 CD FF 26 D8 CD 6B : FF
D488 27 D8 3E 08 37 C0 7E CD : 87
D490 7C 25 D8 36 00 E5 01 1E : B3
D498 00 09 7E E1 CD 4E 27 D8 : 82
D4A0 2A 64 1F 3E 01 CD 5A 25 : 38
D4A8 D4 10 27 C9 3A 5D 1F CD : 57
D4B0 9C 25 D8 CD 91 25 D8 D5 : C9
D4B8 CD 6B 27 ED 53 DF 27 22 : C7
D4C0 E1 27 D1 D8 3E 08 37 C0 : EE
D4C8 7E CD 7C 25 D8 3A 5D 1F : 7A
D4D0 F5 CD A3 1F F1 32 5D 1F : 23
D4D8 CD 6B 27 D8 3E 0A 37 C8 : 7E
D4E0 ED 5B DF 27 2A 64 1F 3E : 39
D4E8 01 CD 44 25 D8 2A 74 1F : CC
D4F0 23 ED 5B E1 27 13 01 11 : 98
D4F8 00 ED B0 ED 5B DF 27 2A : 15
---------------------------------
SUM: 61 10 EB ED 12 F7 CE 75 AC1F

D500 64 1F 3E 01 CD 5A 25 C9 : D7
D508 3A 5D 1F CD 9C 25 D8 CD : E9
D510 91 25 D8 CD 6B 27 D8 3E : 03
D518 08 37 C0 CB F6 2A 64 1F : 6D
D520 3E 01 CD 5A 25 C9 3A 5D : EB
D528 1F CD 9C 25 D8 CD 91 25 : 08
D530 D8 CD 6B 27 D8 3E 08 37 : 8C
D538 C0 CB B6 2A 64 1F 3E 01 : 2D
D540 CD 5A 25 C9 08 3A 5D 1F : D3
D548 CD 9C 25 D8 CD 91 25 D8 : C1
D550 D6 41 32 06 2B 08 CD 00 : 4F
D558 2B C9 08 3A 5D 1F CD 9C : 1B
D560 25 D8 CD 91 25 D8 D6 41 : 6F
D568 32 06 2B 08 CD 03 2B C9 : 2F
D570 F5 3E 01 18 02 F5 AF 32 : 24
D578 1E 29 F1 C9 B7 CB 77 C8 : C2
---------------------------------
SUM: 31 83 ED 91 0B 50 8D 44 6943

D580 3E 04 37 C9 E5 E6 87 21 : B5
D588 1F 29 BE E1 C8 3E 06 37 : 2A
D590 C9 FE 41 38 04 FE 45 3F : C6
D598 D0 3E 0B C9 CD 51 28 D8 : 00
D5A0 CD 63 28 20 04 3E 03 37 : F4
D5A8 C9 CD 91 25 C9 3A 20 29 : 98
D5B0 CD 63 28 C0 3A 7D 1F B7 : A5
D5B8 20 02 3E 54 FE 01 20 02 : D5
D5C0 3E 53 FE 03 20 02 3E 51 : 43
D5C8 C9 F5 32 20 29 FE 54 20 : AB
D5D0 01 AF FE 53 20 02 3E 01 : 62
D5D8 FE 51 20 02 3E 03 32 7D : 61
D5E0 1F F1 C9 2A 74 1F 01 1E : B5
D5E8 00 09 7E 32 DE 27 ED 4B : F6
D5F0 72 1F 2A 70 1F E5 3A DE : 47
D5F8 27 2A 62 1F 5F 16 00 19 : 60
---------------------------------
SUM: 37 89 81 67 FA AF 86 D7 CB4A

D600 7E 32 DE 27 EB 29 29 29 : 1B
D608 29 EB E1 B7 28 19 FE 80 : 6B
D610 30 19 3E 10 CD 44 25 D8 : A5
D618 11 00 10 19 E5 69 60 B7 : 9F
D620 ED 52 4D 44 E1 30 CE 3E : ED
D628 07 37 C9 D6 7F FE 11 30 : 9B
D630 F6 3D 0B B8 20 F1 06 00 : 0D
D638 03 B7 28 07 F5 CD 44 25 : 14
D640 38 14 F1 D5 1E 00 57 19 : A0
D648 E3 5F 16 00 19 EB 2A 64 : EA
D650 1F 3E 01 CD 44 25 D1 D8 : 3D
D658 ED B0 AF C9 ED 5B DF 27 : 63
D660 2A E1 27 ED 4B 72 1F C5 : C0
D668 0B CB 38 CB 38 CB 38 CB : DF
D670 38 04 CD 21 27 B8 C1 3E : 08
D678 09 D8 2A 74 1F E5 D5 C5 : 1D
---------------------------------
SUM: 72 9C 63 98 6B 20 F3 DA 925E

D680 11 18 00 19 5D 54 13 36 : 3C
D688 00 01 07 00 ED B0 C1 D1 : 37
D690 E1 3E 1E 85 6F 30 01 24 : 86
D698 CD 36 27 77 2A 70 1F E5 : 3F
D6A0 2A 62 1F 5F 16 00 19 EB : 24
D6A8 29 29 29 29 EB 0B 78 03 : 15
D6B0 FE 10 38 21 36 80 CD 36 : 20
D6B8 27 77 E1 F5 3E 10 CD 5A : E9
D6C0 25 38 10 11 00 10 19 E5 : 8C
D6C8 69 60 B7 ED 52 4D 44 E1 : 31
D6D0 F1 18 CC E1 C9 3C F5 C6 : 76
D6D8 7F 77 F1 E1 CD 5A 25 D8 : EC
D6E0 CD 10 27 D8 2A 74 1F ED : 86
D6E8 5B E1 27 01 20 00 ED B0 : 21
D6F0 2A 64 1F ED 5B DF 27 3E : 39
D6F8 01 CD 5A 25 D8 AF C9 D5 : 72
---------------------------------
SUM: 88 E8 F8 5E BD 34 92 A2 ACA9

D700 E5 ED 5B 5E 1F 2A 62 1F : 55
D708 3E 01 CD 44 25 E1 D1 C9 : F0
D710 D5 E5 ED 5B 5E 1F 2A 62 : 0B
D718 1F 3E 01 CD 5A 25 E1 D1 : 5C
D720 C9 C5 E5 06 80 0E 00 2A : 31
D728 62 1F 7E B7 20 01 0C 23 : 06
D730 10 F8 79 E1 C1 C9 C5 E5 : 96
D738 06 80 2A 62 1F 7E B7 28 : 8E
D740 06 23 10 F9 37 18 04 3E : C3
D748 80 90 B7 E1 C1 C9 D5 E5 : EC
D750 ED 5B 62 1F 6F 26 00 19 : 77
D758 7E 36 00 FE 80 38 F5 E1 : 40
D760 D1 FE 90 30 02 AF C9 3E : 47
D768 07 37 C9 C5 0E 10 ED 5B : 32
D770 60 1F 2A 64 1F 3E 01 CD : 38
D778 44 25 38 24 06 08 7E FE : 4F
---------------------------------
```

▶こんにちは，初投稿の川田路之（通称，歩くまゆ毛）です。なにかと迷惑をかけることがあるかもしれませんが，常連目指してガンバルつもりです。などと腰を低くしているのもいまのうちだー。ふっふっふっ，「Oh!X征服」のためビシバシ投稿するぜ。

川田　路之　(15)　鹿児島県

```
SUM: C5 2A 00 3E 98 E9 C9 F6 276E

D780 FF 28 1A B7 28 0B D5 ED : ED
D788 5B 74 1F CD CD 27 D1 28 : A8
D790 0D D5 11 20 00 19 D1 10 : 0D
D798 E5 13 0D 20 D5 3E AF B7 : 9E
D7A0 C1 C9 C5 0E 10 ED 5B 60 : 15
D7A8 1F 2A 64 1F 3E 01 CD 44 : 1C
D7B0 25 38 16 06 08 7E B7 28 : DE
D7B8 11 FE FF 28 0D D5 11 20 : 49
D7C0 00 19 D1 10 F0 13 0D 20 : 2A
D7C8 E0 3E AF C1 C9 C5 D5 E5 : D6
D7D0 06 10 13 23 1A BE 20 02 : 46
D7D8 10 F8 E1 D1 C1 C9 00 00 : 44
D7E0 00 00 00 C5 D5 E5 ED 5B : C7
D7E8 74 1F 01 20 00 ED B0 CD : 1E
D7F0 27 28 3A 5D 1F CD F4 1F : E5
D7F8 3E 3A CD F4 1F CD 9D 1F : E1
---------------------------------
SUM: 31 8D 11 1A D4 95 46 35 99ED

D800 CD 2A 29 ED 4B 72 1F 2A : 13
D808 70 1F ED 5B 6E 1F CD 1E : 4F
D810 28 09 2B CD 1E 28 EB CD : 27
D818 1E 28 E1 D1 C1 C9 3E 3A : FA
D820 CD F4 1F CD BE 1F C9 F5 : 48
D828 11 A9 28 CB 7F 28 03 3E : 95
D830 08 11 E6 07 6F 26 00 29 : C4
D838 29 11 A9 28 19 EB CD E5 : C1
D840 1F F1 CB 77 3E 2A 20 02 : DC
D848 3E 20 CD F4 1F CD F1 1F : 1B
D850 C9 CD 63 28 C8 FE 41 38 : 60
D858 07 FE 4D 3F 38 02 B7 C9 : 4B
D860 3E 03 C9 FE 54 C8 FE 53 : 75
D868 C8 FE 51 C9 3D FE 0E 38 : 61
D870 13 3C 11 E3 2A F5 CD E8 : 17
D878 1F 3E 24 CD F4 1F F1 CD : 1F
---------------------------------
SUM: F7 90 8F F6 69 AB 81 F2 52E3

D880 C1 1F 18 0E 21 00 2A 87 : D8
D888 5F 16 00 19 5E 23 56 CD : 32
D890 E8 1F CD C4 1F CD EB 1F : 8E
D898 C9 20 43 6C 75 73 74 65 : 59
D8A0 72 73 20 46 72 65 65 0D : 94
D8A8 00 4E 75 6C 00 42 69 6E : 48
D8B0 00 42 61 73 00 3F 3F 3F : D3
D8B8 00 41 73 63 00 3F 3F 3F : D4
D8C0 00 3F 3F 3F 00 3F 3F 3F : 7A
D8C8 00 44 69 72 00 00 00 00 : 1F
D8D0 00 00 00 00 00 00 00 00 : 00
D8D8 00 00 00 00 00 00 00 00 : 00
D8E0 00 00 00 00 00 00 00 00 : 00
D8E8 00 00 00 00 00 00 00 00 : 00
D8F0 00 00 00 00 00 00 00 00 : 00
D8F8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 43 3B 39 90 85 C7 6A 10 3F2B

D900 C3 32 1C C3 18 1B C3 28 : F2
D908 1C C3 2D 1C C3 37 1C C3 : 01
D910 73 1B C3 E3 27 C3 51 28 : 97
D918 C3 63 28 00 00 00 00 00 : 4E
D920 41 00 00 00 00 00 00 00 : 41
D928 00 00 E5 2A 92 14 22 72 : 49
D930 1F 2A 94 14 22 70 1F 2A : CC
D938 96 14 22 6E 1F E1 C9 E5 : E8
D940 2A 72 1F 22 92 14 2A 6E : 1B
D948 1F 22 96 14 2A 70 1F 22 : C6
D950 94 14 E1 C9             : 52
---------------------------------
SUM: E8 59 65 6D 91 FE 83 24 C045


DA00 1C 2A 2D 2A 3C 2A 50 2A : 7D
DA08 60 2A 6B 2A 79 2A 8E 2A : 7A
DA10 9D 2A A9 2A BD 2A CE 2A : 79
DA18 DC 2A EA 2A 44 65 76 69 : A2
DA20 63 65 20 49 2F 4F 20 45 : 14
DA28 72 72 6F 72 0D 44 65 76 : F1
DA30 69 63 65 20 4F 66 66 6C : D8
DA38 69 6E 65 0D 42 61 64 20 : 70
DA40 46 69 6C 65 20 44 65 73 : BC
DA48 63 72 69 70 74 65 72 0D : 06
DA50 57 72 69 74 65 20 50 72 : ED
DA58 6F 74 65 63 74 65 64 0D : F5
DA60 42 61 64 20 52 65 63 6F : B0
DA68 72 64 0D 42 61 64 20 46 : 50
DA70 69 6C 65 20 4D 6F 64 65 : DF
DA78 0D 42 61 64 20 41 6C 6C : 4D
---------------------------------
SUM: 35 84 5E 22 10 E4 4F B3 FD76

DA80 6F 63 61 74 69 6F 6E 20 : 0D
DA88 54 61 62 6C 65 0D 46 69 : A4
DA90 6C 65 20 6E 6F 74 20 46 : A8
DA98 6F 75 6E 64 0D 44 65 76 : E2
DAA0 69 63 65 20 46 75 6C 6C : E4
DAA8 0D 46 69 6C 65 20 41 6C : 5A
DAB0 72 65 61 64 79 20 45 78 : F2
DAB8 69 73 74 73 0D 52 65 73 : FA
DAC0 65 72 76 65 64 20 46 65 : E1
DAC8 61 74 75 72 65 0D 46 69 : DD
DAD0 6C 65 20 6E 6F 74 20 4F : B1
DAD8 70 65 6E 0D 53 79 6E 74 : FE
DAE0 61 78 20 45 72 72 6F 72 : 03
DAE8 20 0D 42 61 64 20 44 61 : F9
DAF0 74 61 0D 43 6F 6D 70 6C : DD
DAF8 65 74 65 20 21 0D 00 00 : 8C
---------------------------------
SUM: EB 29 41 70 6C 61 CD D8 84F5

DB00 C3 07 2B C3 5E 2B 00 C5 : 06
DB08 D5 E5 CD 11 2B E1 D1 C1 : 36
DB10 C9 ED 73 D7 2C CD B8 2B : DC
DB18 CD DC 2B CD 0F 2C 3E 05 : 1F
DB20 F5 E5 3E 80 CD 69 2C D5 : CF
DB28 11 FB F8 4B ED 78 4A ED : EB
DB30 78 0F 30 0B 0F 30 F8 4B : 44
DB38 ED 78 77 23 4A 18 F0 E6 : 37
DB40 4E D1 28 0B E1 F1 3D CA : 2B
DB48 CA 2C CD 88 2C 18 D1 F1 : 51
DB50 F1 08 3D CA 79 2C 08 CD : 7A
DB58 A5 2C 30 C2 18 BD C5 D5 : 32
DB60 E5 CD 68 2B E1 D1 C1 C9 : 81
DB68 ED 73 D7 2C CD B8 2B CD : E0
DB70 DC 2B CD 0F 2C 3E 05 F5 : 47
DB78 E5 3E A0 CD 69 2C D5 11 : 0B
---------------------------------
SUM: DA F6 81 C3 B8 13 C6 A2 2E3C

DB80 FB F8 ED 78 0F 30 0B 0F : B1
DB88 30 F8 7E 4B ED 79 23 4A : C4
DB90 18 F0 CB 6F C2 C7 2C E6 : DD
DB98 7E D1 28 0D E1 F1 3D 20 : B3
DBA0 03 C3 CA 2C CD 88 2C 18 : 55
DBA8 CE F1 F1 08 3D CA 79 2C : 64
DBB0 08 CD A5 2C 30 BF 18 BA : 67
DBB8 E5 6F 08 26 04 3E FF 2D : F0
DBC0 95 6F B7 ED 52 E1 DA C1 : 76
DBC8 2C 7B 07 CB 12 07 CB 12 : 6F
DBD0 07 CB 12 07 CB 12 7B E6 : 29
DBD8 0F 3C 5F C9 E5 F3 CD 35 : 4D
DBE0 2C 3A 06 2B E6 03 CB 3A : 85
DBE8 30 02 F6 10 F6 80 32 06 : E6
DBF0 2B 0E FC ED 79 D5 1E 03 : 91
DBF8 21 00 00 01 F8 0F ED 78 : 8E
---------------------------------
SUM: FE DC ED 76 3E 04 48 33 E55F

DC00 E6 81 28 2E 2B 7C B5 20 : 39
DC08 F5 1D 20 F2 C3 C4 2C 0E : E5
DC10 FB ED 51 E5 CD 5A 2C 7E : EF
DC18 0E F9 ED 79 72 E1 0E F8 : C6
DC20 3E 1E ED 79 E5 D5 06 20 : A2
DC28 10 FE 01 F8 0F ED 78 0F : 8A
DC30 38 FB D1 E1 C9 D5 16 FF : 98
DC38 01 FB 0F 3E A5 ED 79 3E : 92
DC40 10 CD 53 2C ED 78 FE A5 : 64
DC48 20 02 D1 C9 15 20 E9 D1 : AB
DC50 C3 C4 2C 3D 2A 34 12 20 : 80
DC58 FA C9 21 D9 2C D5 3A 06 : FE
DC60 2B E6 0F 5F 16 00 19 D1 : 7F
DC68 C9 FB 0E FA F3 ED 59 0E : 13
DC70 F8 ED 79 3E 07 3D 20 FD : FD
DC78 C9 F5 01 FC 0F 3A 06 2B : 35
---------------------------------
SUM: 0D B5 5C AC 06 04 F3 B3 6AA6

DC80 E6 03 ED 79 F1 FB B7 C9 : BB
DC88 F5 E5 D5 CD 5A 2C 36 00 : 38
DC90 AF 01 F9 0F ED 79 0D 3E : 69
DC98 02 ED 79 CD 24 2C D1 CD : 23
DCA0 0F 2C E1 F1 C9 1C 3E 10 : 40
DCA8 BB D0 1E 01 E5 21 06 2B : E1
DCB0 7E EE 10 77 E6 10 20 02 : 0B
DCB8 14 37 0E FC 7E ED 79 E1 : 1A
DCC0 C9 3E 05 01 3E 02 01 3E : 8C
DCC8 04 01 3E 01 CD 79 2C ED : A3
DCD0 7B D7 2C 37 C9          : 7E
---------------------------------
SUM: 30 0D C0 C0 42 81 D5 1D AAB2
```

リスト2 FORMAT & SYSGEN

```
7000 CD E2 1F 0C 31 29 20 4C : A0
7008 6F 67 69 63 61 6C 20 46 : D5
7010 6F 72 6D 61 74 0D 32 29 : 8B
7018 20 26 20 53 79 73 67 65 : 71
7020 6E 0D 33 29 20 45 6E 64 : 0E
7028 20 6F 66 20 57 6F 72 6B : B8
7030 0D 0D 49 6E 70 75 74 20 : 4A
7038 57 6F 72 6B 20 4E 6F 2E : AE
7040 20 00 CD 21 20 FE 33 20 : 7F
7048 06 3E 0C CD F4 1F C9 FE : F7
7050 31 38 EF FE 33 30 EB 32 : D6
7058 5A 72 CD F4 1F CD E2 1F : 7A
7060 0D 0D 44 72 69 76 65 20 : 34
7068 4E 61 6D 65 20 3D 20 00 : FE
7070 CD 21 20 FE 61 38 07 FE : AA
7078 65 D2 70 70 D6 20 FE 41 : 4C
---------------------------------
SUM: FB 22 3F 6A AC B1 EF 0B 9439

7080 DA 70 70 FE 45 D2 70 70 : AF
7088 32 5D 1F CD F4 1F CD E2 : 3D
7090 1F 0D 0D 41 6C 6C 20 52 : C4
7098 69 67 68 74 20 3F 20 20 : 4B
70A0 28 59 2F 4E 29 20 20 00 : 67
70A8 CD 21 20 FE 59 C2 00 70 : 97
70B0 3E 01 11 00 2E 12 13 3E : E1
70B8 8F 12 13 AF 12 21 02 2E : C6
70C0 13 3A 66 1F D6 03 4F 06 : 00
70C8 00 ED B0 3E 8F 23 13 77 : 17
70D0 3A 66 1F 4F 3E FF 91 4F : 2B
70D8 06 00 ED B0 3E 01 ED 5B : 2A
70E0 5E 1F 21 00 2E CD 03 20 : BC
70E8 DA D8 71 21 00 80 3E FF : 01
70F0 77 11 01 80 01 FF 0F ED : 05
70F8 B0 3E 10 ED 5B 60 1F 21 : E6
---------------------------------
SUM: 08 A1 3C 65 F2 83 01 F4 1978

7100 00 80 CD 03 20 DA D8 71 : 93
7108 3A 5A 72 FE 31 20 12 CD : 34
7110 E2 1F 0D 0D 43 6F 6D 70 : AA
7118 6C 65 74 65 20 21 0D 00 : F8
7120 C9 21 02 2E 36 03 23 36 : AC
7128 04 23 36 05 23 36 8F 3E : 88
7130 01 ED 5B 5E 1F 21 00 2E : 15
7138 CD 03 20 DA D8 71 21 00 : 34
7140 00 11 00 80 01 00 30 ED : AF
7148 B0 CD F7 1F 7C FE 20 30 : 5D
7150 14 AF 32 14 2B 32 20 2B : B1
7158 32 EE 2B 7C FE 10 30 05 : 0A
7160 3E 12 32 02 80 3E 30 11 : 83
7168 20 00 21 00 80 CD 03 20 : B1
7170 DA D8 71 3E 01 11 00 00 : 73
7178 21 00 2F CD 00 20 DA D8 : EF
---------------------------------
SUM: 72 F7 BA 1A AB D1 E4 A6 02E3

7180 71 CD F7 1F 7C FE 20 30 : 1E
7188 0E FE 10 38 05 21 FA 71 : E5
7190 18 08 21 1A 72 18 03 21 : 09
7198 3A 72 11 00 2F 01 20 00 : 0D
71A0 ED B0 3E 01 11 00 00 21 : 0E
71A8 00 2F CD 03 20 38 29 CD : 4D
71B0 F7 1F 7C FE 20 30 0F 21 : 10
71B8 14 2B 36 2F 21 20 2B 36 : 46
71C0 2F 21 EE 2B 36 01 CD E2 : 4F
71C8 1F 0D 0D 43 6F 6D 70 6C : 34
71D0 65 74 65 20 21 0D 00 C9 : 55
71D8 CD EE 1F CD 33 20 CD E2 : A9
71E0 1F 52 45 54 52 59 20 28 : FD
71E8 59 2F 4E 29 20 3F 20 20 : 9E
71F0 00 CD 21 20 FE 59 CA 00 : 2F
71F8 70 C9 01 49 50 4C 50 52 : C1
---------------------------------
SUM: 31 15 2A E3 4D 98 04 9A E01F

7200 4F 53 2D 4F 53 20 53 57 : 3B
7208 4F 52 44 0D 00 00 00 30 : 22
7210 00 00 00 00 00 00 00 FF : FF
7218 20 00 03 49 50 4C 50 52 : AA
7220 4F 53 2D 4F 53 20 53 57 : 3B
7228 4F 52 44 0D 00 00 00 30 : 22
7230 00 00 00 00 00 00 00 FF : FF
7238 20 00 01 53 2D 4F 53 20 : 63
7240 53 57 4F 52 44 20 20 20 : EF
7248 53 79 73 20 00 30 00 00 : 8F
7250 00 00 00 00 00 00 00 00 : 00
7258 20 00 00 00             : 20
---------------------------------
SUM: 42 1A A8 C6 67 2B 69 9E 54CD
```

●X1用ロールプレイングゲーム

FLAME

Terakawa Makoto 寺川 誠

X1用にオールBASICのRPGが登場だ。「炎」の反乱に立ち向かい，ICE SWORDを手に入れて地獄のともしびを打ち破れ。適度な広さのマップとほどよい謎解きと，なかなかバランスのよいゲームだ。要所をしっかり押さえていけば比較的簡単にクリアできるぞ。

まずはストーリー解説

このプログラムはBASIC CZ-8FB01を使ったロールプレイングゲームです。ダンジョンとなっている塔の中を昇ったり降りたりしながら，情報を集めアイテムを揃えていってください。

タイトルのFLAMEとは炎を意味する言葉です。そう，それはこのゲームでの敵を表す言葉です。そして，そもそもの発端は人と炎の関係までさかのぼります。それでは，このゲームのバックグラウンドストーリーから始めてみましょう。

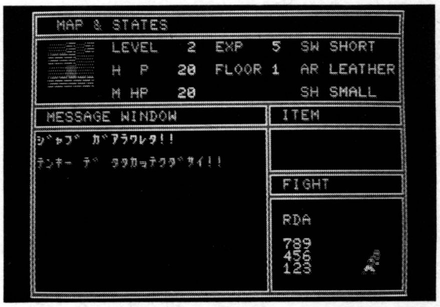
それはいまからずっとずっと昔の話，まだ火を知らず，原始的な生活をしている村がありました。そのころは「炎」は万物を焼き尽くす魔物のように思われていたのです。人々は炎の力を恐れ，超自然的な存在として炎をあがめていました。

ふとしたことから，ひとりの若者が炎を自由に操れるようになりました。炎の力を利用することによって一挙に文明は花開き，国は栄えました。しかし，時がたつにつれて，宝物のようにあがめられていた炎は次第に粗末に扱われるようになっていったのです。やっと人々に気にいられたのにと思った炎はこういった事態に不満を持ち始めました。そして「地獄のともしび」となって数多くの魔物たちを集めて「ラーヴァの塔」にたてこもったのです。

突然，村から炎が消えてしまい，炎のある生活に慣れきっていた人々は混乱の極みにありました。そんなときです。ラーヴァの塔に閉じ込められていたうちのひとりがやっとのことで，村に逃げ帰ってきました。村の人々は炎が地獄のともしびに変わってしまったことを知りました。もはや炎の力がなくては人々の生活は成り立ちません。人々は己の浅はかさを恥じました。そして，村を救うためひとりの若者がラーヴァの塔に向かったのです。

入力&操作方法

このプログラムは2つのプログラムから成り立っています。リスト1はPCGデータやモンスター，迷路などのデータなどの集まりです。リスト2はこのゲームを動かしているメインプログラムにあたります。



それではBASIC CZ-8FB01（Ver.1.0）を立ち上げ，リスト1とリスト2を打ち込んでください。特にリスト1はDATA文の山なので入力ミスのないように。両方のプログラムをデフォルトドライブにセーブしておき（リスト2のファイル名は必ず“FLAME MAIN”とすること），リスト1を起動するとゲームが始まります。PCGのセッティングとマップの初期化に少々時間がかかりますが気長に待ってください。タイトル表示のあと「PUSH SPACE OR TRIGGER」と表示されるので，スペースキーかジョイスティックのトリガーボタンを押してください。これによって入力デバイスが選択されます。ただし，移動などの操作ではジョイスティックも使用できますが，戦闘モード（後述）ではテンキーしか使えないので注意が必要です。

ゲーム解説

さて，画面構成は左上に5×5エリアのマップ（いつも自分が中心にいる），その右側がキャラクターのステータスを表示します。ステータスはLEVEL，HP（現在のヒットポイント），MHP（ヒットポイントの上限），EXP（経験値），FLOOR（塔の何階にいるか），SW（装備している剣），AR（装備している鎧），SH（装備している盾）を表しています。ステータスの下側には現在持っているアイテムが表示されます。画面左下はメッセージウィンドウで，メッセージやゲームキャラクターとの会話，そのほ

表1 変数表

MO$( )	モンスターの名前	NL	次のLEVELまでの経験
MH( )	モンスターのHP	ST	ジョイスティックの番号
ME( )	モンスターのEXP	SA	KEYの番号
MT( )	モンスターのタイプ	PR	メッセージの表示に使用
ME$( )	メッセージ	PR$	メッセージの表示に使用
M$( )	マップ	TE	モンスターの番号
MA$( )	マップの表示に使用	KO	自分の行動
MX	自分のX座標	HI	自分の行動の高さ
MY	自分のY座標	AP	自分の攻撃値
SW	SWORDの強さ	DP	自分の防御値
SH	SHIELDの強さ	KT	モンスターの行動
AR	ARMORの強さ	HT	モンスターの行動の高さ
HP	自分のHP	TA	モンスターの攻撃値
MH	HPの上限	TH	モンスターのHP
LV	LEVEL	AD	アドレス
EX	経験値	SW$	SWORDの名前
FI	階段のフラグ	SH$	SHIELDの名前
F2	会話のフラグ	AR$	ARMORの名前
F3	ITEMのフラグ		
FL	現在いるFLOOR		
FF	FIGHTのフラグ		

その他，1文字の変数はいろいろなところで使用しています。

かの情報が表示されます。右段下側は戦闘モード用のウィンドウです。

キー操作は移動モードで，テンキーまたはジョイスティックの4方向，あたりを調べたいときにはスペースキーかトリガーを押してください。戦闘モードではテンキーのみになります。戦闘モードはちょっと複雑なので詳しく解説しておきましょう。

画面右下のウィンドウを見てください。

R	D	A
7	8	9
4	5	6
1	2	3

と表示されていますね。これはRUN（戦闘回避）なら1，4，7のうちのどれか，DEFENCE（防御）なら2，5，8のうちのどれか，ATTACK（攻撃）なら3，6，9のうちのどれかを押すということです。3つの数字は上から順に頭，胸，腹を意味します。たとえば，敵の頭を攻撃する場合なら9となります。相手も同様に行動しますから，その組み合わせで攻撃の成否が決定します。このときの判定は，

剣対剣	同じ部分で攻撃失敗
	違う部分で相打ち
盾対剣	同じ部分で防御成功
	違う部分で防御失敗
盾対盾	防御成功
回避	同じ部分で回避失敗
	違う部分で回避成功

となっています（回避のときの相手はすべて同じ）。

戦闘に勝つとEXPがあがり，現在のレベル×30のEXPに達するとレベルアップしていきます。レベルがあがるとヒットポイントの上限があがり，ヒットポイントも回復します。このときEXPは0に戻されます。

情報集め

スペースキー（またはトリガー）を押すと現在自分がいる場所を調べることができます。また，人と会話をするときも同様にスペースキーを押してください。このゲームでは情報集めが重要ですから，マップ上に変なものがあったり，生き残った人をみつけたときには忘れずに情報を集めてください。たとえ，なにかがありそうでも，するべきことをしていないと取るべきものも取れないことがあります。

なお，キャラクターのロード/セーブは最初にスタートした地点を調べることで行われます。ロード/セーブは何回でもできますが，ロード直後にはEXPが0になっていますので注意してください。

アイテム類は行き止まりによく落ちています。そして，なにもないのに調べるとヒットポイントを回復してくれる場所が何カ所かあります。いくらでも使えて各フロア（ラーヴァの塔は5階建て）に必ずあるのでしっかりみつけて使ってください。順序よくやれば1時間くらいで解けるのではないかと思います。それでは皆さんがんばってください。

Profile

◇寺川さんは福岡県にお住まいの17歳，高校2年生です。マイコン歴，約5年のX1turboIIIユーザーです。以前にはショートプログラム，WARP！やTHRILLINGを発表したこともありますね。

リスト1 FLAME DATA

```
10 '┌──────────────────────────────────────┐
20 '│ F L A M E  BY MAKOTO TERAKAWA │
30 '│            For X1/T (CZ8FB01) │
40 '│            ROLE PLAYING GAME  │
50 '└──────────────────────────────────────┘
60 '────── S T A R T ──────
70 CONSOLE 0,25:WIDTH 40:SCREEN 0,0:CLEAR &HF500:PRW &B11111110
80 FOR I=1 TO 7:PALET I,0:NEXT:KBUF OFF:CLICK OFF:CGEN:COLOR 7
90 DIM M$(21),ME$(25),MO$(11),MH(11),ME(11),MT(11),MA$(5)
100 '────── P C G  S E T ──────
110 LOCATE 5,10:PRINT"NOW PCG SETTING. PLEASE WAIT."
120 S=32:E=90:GOSUB 130:S=160:E=223:GOSUB 130:GOTO 160
130 FOR I=S TO E:A$=LEFT$(CGPAT$(I),8):B$="":C$="":D=0:FOR E=1 TO 8
140 A=ASC(MID$(A$,E,1)):B=A*2:C=B OR D:D=A:B$=B$+CHR$(B):C$=C$+CHR$(C):NEXT E
150 DEFCHR$(I)=C$+B$+B$:NEXT:RETURN
160 READ A:IF A<>999 THEN READ B$:DEFCHR$(A)=HEXCHR$(B$):GOTO 160
170 '────── M A P  E D I T ──────
180 LOCATE 5,10:PRINT"NOW MAP SETTING. PLEASE WAIT."
190 A$="                    ":M$(0)=A$:M$(21)=A$:AD=&HF690:N=247:AN=0
200 FOR I=1 TO 5:FOR J=1 TO 20:READ A,B:A$=BIN$(A):B$=BIN$(B)
210 M$(J)=STRING$(10-LEN(A$),"0")+A$+STRING$(10-LEN(B$),"0")+B$
220 NEXT J:FOR K=1 TO 20:FOR L=1 TO 20
230 IF MID$(M$(K),L,1)="0" THEN POKE AD+AN,224 ELSE POKE AD+AN,N
240 AN=AN+1:NEXT L,K
250 READ C:IF C<>999 THEN READ D:POKE AD+C,D:GOTO 250
260 AN=0:AD=AD+400:N=N+1:NEXT I
270 FOR I=0 TO 9:READA$,B$:FOR Q=1 TO 20:POKE &HF500+Q+I*20-1,ASC(MID$(A$,Q,1))
280 POKE &HF67C+Q-I*20-1,ASC(MID$(B$,Q,1)):NEXT Q,I:POKE &HF641,246
290 '────── MESSAGE & MONSTER  READ ──────
300 FOR I=1 TO 25:READ ME$(I):NEXT
310 FOR I=1 TO 11:READ MO$(I),MH(I),ME(I),MT(I):NEXT I
320 '────── T I T L E ──────
330 C=0:CLS:CGEN 1:LOCATE 0,12:COLOR 5:FOR I=23 TO 72:IF C=-1 THEN C=7
340 LINE(0,I)-(319,I),PSET,C:C=C-1:NEXT
350 PRINT"    R O L E  P L A Y I N G  G A M E    ":PRINT:PRINT:PRINT:COLOR 7
360 PRINT"          BY  MAKOTO TERAKAWA           ":LOCATE 0,3
370 SOUND 7,&B111100:PLAY"V15O4C2CCED4C2ED4C2ED4+C2+C+C4C2ED4-C2-C:V13O4R0C2CCED
4C2ED4C2ED4R4RC2ED4-C2-C"
380 PRINT"  ±±±±±± ±±±      ±±±   ±±±   ± ±±±±±± "
390 PRINT"  ±±±    ±±±     ±±± ±  ±±±± ±± ±±±    "
400 PRINT"  ±±±    ±±±    ±±±   ± ±±± ± ± ±±±±±± "
410 PRINT"  ±±±±±± ±±±    ±±±   ± ±±±   ± ±±±    "
420 PRINT"  ±±±    ±±±    ±±± ±±± ±±±   ± ±±±    "
430 PRINT"  ±±±    ±±±±±± ±±±   ± ±±±   ± ±±±±±± "
440 B=2:FOR I=1 TO 6:A=256-B:B=B*2:PRW A:PAUSE1:NEXT:PRW 0
450 REPEAT:COLOR INT(RND*7)+1:LOCATE 8,21:PRINT "PUSH SPACE OR TRIGGER"
460 A=STRIG(0):B=STRIG(1):UNTIL A<>0 OR B<>0
470 J=64:FOR I=1 TO 6:K=256-J:J=J/2:PRW K:PAUSE1:NEXT
480 CLS 4:PALET:PRW &B11111100:IF A=-1 THEN ST=0 ELSE ST=1
490 '────── SCREEN SET ──────
500 FOR I=1 TO 9:READ A,B,C,D:LINE(A,B)-(C,D),PSET,7,B:NEXT:PAINT(3,3),1,7
510 FOR I=1 TO 16:READ A,B,C,D$:LOCATE A,B:COLOR C:PRINT D$:NEXT
```

```
520 LINE(32,40)-(40,47),PSET,1,BF:CHAIN "FLAME MAIN"
530 '———— P C G  D A T A ————
540 DATA  91,101818181800181800080808087C383810383838387C0000
550 DATA  92,000000000000000000000000000000000E0000000000071F71
560 DATA  93,000000000000000000000000000000000000000000E0F8FE
570 DATA  94,0E1F39393F0E00001F3F39393F0E0000EEDFF9F9FF7F1F07
580 DATA  95,00314B4E34000000317A4D4A34000000CEB4C8C9FBFEF8E0
590 DATA  96,0000000000000000000000000103070D0000000000000101
600 DATA  97,000000201C0000003C7E7E5EE2BEDCE00000140000008080
610 DATA  98,00000C1F3F19A0F00A050F1F3F39400000000C1F3F190000
620 DATA  99,00000080C0E0A0F0ECDCA080C0E040000C0C0080C0E00000
630 DATA 100,0000000202000000103171E0E071F3F0103000202000000
640 DATA 101,000000808007020000800D0F0E0C0E0FB00800080800060E0
650 DATA 102,0000000000000000507F7FC0DF1F3E3C2C031F1F1F07000050
660 DATA 103,000202070707070717FFFC00F0F0F87868E0F0F0F0C0000014
670 DATA 104,00000000000003070500150055075C78071F3F7FFFF8E000
680 DATA 105,00000000000080C04000500054C0743CC0F0F8FCFE3E0E00
690 DATA 106,0503000000030F033A0C00000000000000020000000000000
700 DATA 107,408000000080E080B8600000000000000080000000000000
710 DATA 108,0206000C1F1E0C000206003060E1B3C701091F336DEDB3C7
720 DATA 109,002070E0F0F06000002070980C0E9AC680C080186C6E9AC6
730 DATA 110,00040E063020027970340E06302002017F3B01190F183978
740 DATA 111,60600040C404048C7C780040C00000809C98E0A01C3CEC4C
750 DATA 112,050A170E1F0F1508070F18101018DFDF070F1F1E1F1FDFD8
760 DATA 113,0080C0E0C2A25C2080C063232262FCF080C0E3E3E0E0E030
770 DATA 114,944A3502010102009F4F3F070101020C1C0F0F070100000C
780 DATA 115,54AA55AA54100C00FEFFFFFFFE100E027EFFFFFFFE000202
790 DATA 116,0103070E1D3B53ED0103060C1970201001030 70E1C3B53ED
800 DATA 117,808040E0E0E0FFFF80000040E0407F41808040E020E0FFCB
810 DATA 118,5E3707030D1E1C3E04020502091010205E3707030D1E1C3E
820 DATA 119,7FFFFFFEBC783C00414141221C00000057CBD7EABC281400
830 DATA 120,00000000000000000000000C06231109060F3F73795C2E76
840 DATA 121,000000000000060B0000000889AF2FCBCF0F8FE77650D62B3
850 DATA 122,0000000000000007F0F112101030D3380F0EEDE76000000
860 DATA 123,6000F0909010105EF8FD0AF8F8FEFD7F6702F5979610105E
870 DATA 124,00000700002020E000304740002C25E4F9C8B8BEFF0F07CE
880 DATA 125,000000000000000000000000103141810140803C7EF7C38307
890 DATA 126,70000000000000007000000700000000720007 0F1C3ACF40
900 DATA 127,00000000071C0000020408E0071C00000E1CF8E0071C807F
910 DATA 128,000000000000000100000000000000010000000000000001
920 DATA 129,07070F0F0F1CDFBF00000000000000DFBF000000010300DFBF
930 DATA 130,80F8FFFFFFFF7FBF000000000000000800000080F8FF7FBF
940 DATA 131,000080F8FFFFFFFE00000000000000000000000000080F8F0
950 DATA 132,010101000000000001010100010303070101010001030307
960 DATA 133,B7BFBF3F1F000000B7BFBFFFDFE0FBFFB7BFBFFFDFE0FBF3
970 DATA 134,BFBFBFBF7F7F3F3F808080800080C0C0BFBFBFBF0780C0F8
980 DATA 135,FEFEFCFCFCF8F8F80000000000000000F0F0E0E0E0400000
990 DATA 136,00000000020202 06070E0C0901010101070E0C0901010101
1000 DATA 137,00000000000000103FF7FFFFFFFFFFEFCC7078F0F1F3F7FFF
1010 DATA 138,3F3F3F7F7FFFFFFFC0C0C08080000000FFFFFFFFFFFFFEF8
1020 DATA 139,F0F0F0E0E0E0C0C000000000000000000808080000000000
1030 DATA 140,07070F0F0F1F1F1C0000000000000000000000101000000 00
1040 DATA 141,079FFFFFFFEF08000F8600000000000000FFFEF0800000000
1050 DATA 142,FFFFF8C0000000000000000000000000C000000000000000
1060 DATA 143,C00000000000000000000000000000000000000000000000
1070 DATA 144,000000000000000000000000000001010000000000000000
1080 DATA 145,00000000000000000000000000003878F000000000000000
1090 DATA 146,00000000000000000000008B4F9FBFFFFF00000102000040CC
1100 DATA 147,000000000000000000F0C090F0F0F0F8008000000000206040
1110 DATA 148,000000000000000000000000011119090300000000000000000
1120 DATA 149,00000002070703000F0F9FBDF8F8FCFF0000008A97272300
1130 DATA 150,000000000E1C1800FFFFFFFFF1E3E7FF999000000E1C1900
1140 DATA 151,00000000000000000F9FAFEFCFCFCF8F08020D0001070C000
1150 DATA 152,000000000000000003030101010000000000000000000001
1160 DATA 153,0301000000000000FBF1F0F0F8FF7F1F4341004020008080
1170 DATA 154,1810000000000000D9100000C1E3FFFF1810000000014183
1180 DATA 155,0000000000000000F0F0E0E0E8C8D8D00000000000080000
1190 DATA 156,000000000000000000000000000000000303070F7FE7DF3F
1200 DATA 157,000000000000000070000000000000000000080C0E0E0E0C0
1210 DATA 158,0000000000000000FF7F0000000000000002000603070F1F0F
1220 DATA 159,0000000000000000C0800000000000000000000060F0F8B8D8
1230 DATA 224,0000000000000000005500AA005500AA0000000000000000
1240 DATA 225,0000000000000000005500AA005500AA0000000000000000
1250 DATA 226,004500280045002800100082001000820045002800450028
1260 DATA 227,005500AA005500AA00000000000000005500AA005500AA
1270 DATA 228,005500AA005500AA00000000000000005500AA005500AA
1280 DATA 229,003C7EFF7F3F1F00004120502850 00AA003C7EFF7C381000
1290 DATA 230,00000000000000000550082004100AA0000000000000000
1300 DATA 231,0000000000000000385554BA008300EE3854547CFEFE6CEE
1310 DATA 232,0000000000000000387D54BA008300EE387C547CFEFE6CEE
1320 DATA 233,0000000000000000385554D6388300EE38545454FEFE6CEE
1330 DATA 234,00000044FE7C6C0038556CFEFEFF6CEE38546C38008200EE
1340 DATA 235,0000007C10106C0038557CFEFEFF6CEE38547C82EEEE00EE
1350 DATA 236,0000007C4C206C00385554FEFEFF6CEE385454 00B2DE00EE
1360 DATA 237,00000000C67C6C00385554FE388300EE3854547CFEFE6CEE
1370 DATA 238,00000000C67C6C00387554DE388300EE3874545CFEFE6CEE
1380 DATA 239,3854540 0C66C7C00000100FE389300EE3854547C389200EE
1390 DATA 240,0000000000000000385554FEFE836CEE3854543800FE00EE
1400 DATA 241,0000000000000000183C7E7EFFFF7E3C0000004224000000
1410 DATA 242,0000001800000000005500BA005500AA0000001800000000
1420 DATA 243,0000000000000000005500AA085500AA0000000008100000
1430 DATA 244,0000002000040000005500AA005500AA0000000000000000
1440 DATA 245,0000000008180000005500AA085D00AA0000000008180000
1450 DATA 246,005500A2004500AA00000000000000005500AA005500AA
1460 DATA 247,00000000000000000000000000000000BF00FB00BF00FB00
1470 DATA 248,548A15AA15AA552A00000000000000000020404040000000
1480 DATA 249,54A255AA15AA512A7EF7FFFFBFFFFB7E0000000000000000
1490 DATA 250,0000000000000000FFEDBFFFEDFFBBFF0000000000000000
```

```
1500 DATA 251,00000000000003C183E7C7EFFFF7E0010220400044280400
1510 DATA 252,0000000000000000BF00FB00BF00FB00000000000000000000
1520 DATA 253,BF00FB00BF00FB000000000000000000BF00FB00BF00FB00
1530 DATA 254,FFFFFFFFFFFFFFFF04114000208802000C33C00030CC0300
1540 DATA 255,FFFFC3FFFF7E3C18FF81BD8181422418FF9783978F562C18,999
1550 '―――――― M A P  D A T A ――――――
1560 DATA 1023,1023, 513,  21, 989, 501, 593, 261, 727, 469, 529,  81,1021, 863
1570 DATA  580, 321, 853, 381, 593, 261, 723, 853, 534,  85,1010, 981, 519, 533
1580 DATA  740, 149, 647, 721, 658,  95, 767, 965, 512,   5,1023,1023
1590 DATA   21,  83,  26, 231,  34, 225,  38, 242, 141, 232, 190, 225, 288, 233
1600 DATA  323, 229, 358, 230, 378, 229, 999
1610 DATA 1023,1023, 520, 129, 747,1023, 546, 137, 958, 545, 512,1021,1022, 529
1620 DATA  514,  87, 736, 469, 687, 277, 644, 381, 756, 833, 644, 607, 956, 721
1630 DATA  544, 661, 751, 693, 674, 133, 698, 959, 546,  17,1023,1023
1640 DATA   31, 235,  38, 229, 144, 236, 196, 225, 212, 234, 323, 230, 338, 243
1650 DATA  365, 225, 376, 229, 378, 230, 999
1660 DATA 1023,1023, 648,  65, 683, 861, 672, 325, 570, 469, 995,  21, 537, 341
1670 DATA  657, 373, 733, 321, 517, 863,1012,  65, 775, 637, 628,   5, 709, 989
1680 DATA  668, 529, 704, 759, 751, 645, 674, 765, 552, 529,1023,1023
1690 DATA   21, 225,  38, 230,  72, 229, 170, 237, 283, 238, 333, 225, 338, 229
1700 DATA  343, 244, 376, 230, 999
1710 DATA 1023,1023, 545, 257,1005, 381, 549, 325, 645, 469, 765,  81, 529, 351
1720 DATA  991, 273, 513, 501, 765,   5, 517,1021,1012,   1, 535,1023, 656,  81
1730 DATA  725, 277, 645, 469,1021, 341, 577, 343, 533,  65,1023,1023
1740 DATA   21, 245,  23, 229,  30, 225,  72, 230,  98, 239, 127, 229, 193, 229
1750 DATA  274, 229, 301, 229, 332, 229, 338, 230, 361, 229, 378, 225, 999
1760 DATA 1023,1023, 513,   1,1017, 511, 569, 273, 675, 341, 685, 341, 681,  69
1770 DATA  687,1021, 673,  53, 669, 901, 725, 125, 669, 261, 705, 501, 575, 277
1780 DATA  896, 469, 767, 885, 512, 261,1023,1021, 512,   1,1023,1023
1790 DATA   23, 230,  38, 241, 108, 225, 127, 230, 193, 230, 206, 254, 272, 225
1800 DATA  274, 230, 301, 230, 332, 230, 361, 230, 999
1810 DATA "円円円円人人人人人人人人人人人人人円円円"
1820 DATA "人人人人人人人円円円円円円円円円円円円"
1830 DATA "円♠♠♠円人♠♠♠♠◆◆◆◆人人円円円円"
1840 DATA "人◆◆◆人♠♠♠●●円●●●▒●●●●円"
1850 DATA "円♠円♠円円♠人♠◆◆◆◆◆◆人●●●円円"
1860 DATA "人◆◆◆人♠円円円●円●円円生円円円♣円"
1870 DATA "円♠円♠♠♠♠人人人人人人人●円●円円"
1880 DATA "人♥◆♠人♠円●●●円●円生生円●円円円"
1890 DATA "円♠円円円人人人人人人●●●●●円●円円"
1900 DATA "人◆♠♠人●円●円円円●円生生円●●●円"
1910 DATA "円♠円♠♠♠♠♠♠♠♠♠●円円円円円●円円"
1920 DATA "人人人♠円●円●●●円●円円円円●円●円"
1930 DATA "円♠♠♠円円円円円円円●円●●●円●円円"
1940 DATA "人♠♠♠円●円円円●円●●●●●●●円●円"
1950 DATA "円円円円円●●●●●円●円●円●円●円円"
1960 DATA "人♠円円円●●●円●円円円円円円円円●円"
1970 DATA "円●●●円●円円円●円●円●円●円●円円"
1980 DATA "円♠円●円円円●円●●●円●●●●●●●円"
1990 DATA "円♠円●円●円●●●円●円●円●●●円円"
2000 DATA "円♠円○●●円●円円円●円●円円円円円円"
2010 '―――――― MESSAGE DATA ――――――
2020 ' 「"」ト「"」ノ チガイ ニ チュウイ!! 「"」ハ CHR$(&H22) デス。 KEY 3,CHR$(&H22) トイウフウニシヨウ
2030 DATA 」ササヌミヤハチツキ゜クト・ヨエズ゜゛ハラエセキ゜カキスウ「カキスニサナアウートウレ「、
2040 DATA 」ラウスマ゜キ゛ソーヒ゛ミケワ「ヒリマワフロヒロ「、
2050 DATA 」キロヌイーチハキ@!チスキサハナエヌ!」スッエク!ハ!サニ、!ナウエスワウサニキ゜イレヒセ゜ニ゛チ゜コナ゜/////「、
2060 DATA 」"@!ンチスヒチ゜ロ@!イイ・クムキ゜チセコトケロチハキ///「カロウヌ!」HSFBU!BSNPS、!アクムヌイコ゜ラエ「、
2070 DATA 」ナ゜エヤ!イルキ゜ナエ「、
2080 DATA 」ナ゜エヤ!イルキ゜ナエ「キニリセ゜!」ス゜サ゜ケ!ハ!ナヤスフ゜、!アチカストケロ「、
2090 DATA 」」リアエ゜ィ、ハウムアスートレキウ@!」ラエキ゜゛、ハサナチ゜ラ「!サカーチヤハ!ア!ナキセヌヒ!」ラエキ゜゛!ハ!サニ、チ゜「、
2100 DATA 」キ゜゛ヒ゜ロラ""、
2110 DATA 」サハナエヌヒ!ツキ!キ゜イレリスウ「スキス!ンチス!ヒ!ウーチサナキ゜ニウ「、
2120 DATA 」サハラサハキホ゜キ゜ス゜ュミト゜!メサエヌウコニウ「」ク゛!ハ!カハ、キ゜イロヒ゜///サハミオ!ミヤハヌ!ナルイコ゜リロチ゛チ゜「、
2130 DATA 」イー"!サロヒ」ク゛!ハ!カハ、チ゜「!イルキ゜ナエ「!カロウヌ!サハ!」MPOH!TXPSE、!アクムヌイコ゜ラエ「、
2140 DATA 」サハイウチ゜!ヒ!ナ゜エヤ!イルキ゜ナエ「、
2150 DATA 」サハミオ!ケツヒ゜ス!ハ!テウチミヤハ!キ゜!」ク゜゛!ハ!テマ゜、ア!エオハマエヌヤートウケハアムチ「、
2160 DATA 」クムヒ@!」ス゜サ゜ケ!ハ!ナヤスフ゜、アチカスヌ@!ウウサナアカスオ//、!キロヒサカルテウチ「ヒユケイチチモニケトヒ/////
2170 DATA ヒユケ"!ヒユケ"
2180 DATA 」ラエキ゜゛!ハ!サニ、アヘルキコレナ・キロハキリチ゜!ヒ!チ゜゛チ゜゛ナコトウーチ「!」エ///ニ//ニ゛チ゜@!イルキ゜ナエ「!カロウヌ
」IZQFS!TIFME、アイコ゜ラエ「!タエタエ」ス゜サ゜ケ!ハ!ナヤスフ゜、!ヒ!JDF!キ゜クリウチ゜「!スキス!JDF!NBHJD!アテキエユテキ゜ウレ
タ゜「、
2190 DATA 」キ゜゛ヒ゜ロラ""、
2200 DATA 」」ク゜゛!ハ!テマ゜、!ヌ!」ソウロウ!ハ!ムセ゜、!ア!ケム!」ス゜サ゜ケ!ハ!ナヤスフ゜、!ヌキコレナ!フヒクオレリスウ「、
2210 DATA 」」ソウロウ!ハ!ムセ゜、!ヒ!ツキヌイレ「ナカヤエコナ゜「、
2220 DATA 」」ク゜゛!ハ!テマ゜、ヒ!サハキウ!ヌ!イレ「ツキ!ヌヤササキリウコレ「!ニソ゜スートレキート///!ヘヘヘ「、
2230 DATA 」ユレ//ニ//!ウウサ/ナ//!ンチス!ハ//テケーチ!JDF!TXPSE!キ゜//!ツキヌ!イレ//イウテ//ヌ/!セトリロチ//、
2240 DATA 」ウロヤハ!キ゜!ニケトヒ!ムセ゜!ヒ!ケモ゛ニ「、
2250 DATA 」シイ!ムセ゜アケムニシウ「ナサワト゜!シウク゛!ツキ!キ゜シメウ゛チ゜「ニヌキカツトウレハキニ@、
2260 DATA 」クムチ゜コキ゜!チラルチ゜「、
2270 DATA 」ラケクチ///シイ!キキートサウ"""「、
2280 '―――――― MONSTER DATA ――――――
2290 DATA NOMAL SLIME ,      5 ,  3 , 1
2300 DATA ジャブ ,           7 ,  5 , 1
2310 DATA オニ ノ コ ,       10 , 10 , 1
2320 DATA ウキガメ ,         15 , 13 , 1
2330 DATA ミハリバン ,       23 , 14 , 1
2340 DATA ゴーゴーラッカセイ ,  30 , 19 , 1
2350 DATA KNIGHT ,          38 , 22 , 1
2360 DATA カミツ キ ,        47 , 23 , 1
2370 DATA ミニドラゴン ,      60 , 28 , 1
2380 DATA トリ ノ タテ ,      80 , 40 , 2
2390 DATA ジゴク ノ トモシビ , 100 ,  0 , 2
2400 '―――――― SCREEN DATA ――――――
2410 DATA 2,2,317,197,5,5,314,18,5,21,314,66,5,69,202,82,5,85,202,194
2420 DATA 205,69,314,82,205,85,314,114,205,117,314,130,205,133,314,194
2430 DATA 3,1,3,MAP & STATES,9,3,5,LEVEL,9,5,5,H  P,9,7,5,M HP,20,3,5,EXP,20
2440 DATA 5,5,FLOOR,29,3,5,SW,29,5,5,AR,29,7,5,SH,2,9,3,MESSAGE WINDOW,27,9,3
2450 DATA ITEM,27,15,3,FIGHT,27,18,5,RDA,27,20,7,789,27,21,7,456,27,22,7,123
```


▶「MZ-700にはフカの脳はない！」，とゆーわけで私がいまではだーれも使っていないといわれる超騒音速プリンタMZ-80P3のオーナーです。愛用している言語は，無敵のBASE-80Ver.3.5。そしてなぜかTK-85があったりするのです。結論「わが家に高解像度CRTはない！」

小林　隆（27）新潟県

リスト2 FLAME MAIN

```
10 '┌─────────────────────────────┐
20 '│ F L A M E  BY  MAKOTO TERAKWA │
30 '│ MAIN       For X1/T (CZ8FB01) │
40 '│   PROGRAM  ROLE PLAYING GAME  │
50 '└─────────────────────────────┘
60 '───────── SETTEI ─────────
70 MX=3:MY=3:F1=1:F2=0:F3=0:FL=1:AD=&HF500:HP=10:MH=10:FF=0:LV=1:EX=0
80 SW=10:AR=5:SH=2:NL=30:SW$="SHORT  ":AR$="LEATHER":SH$="SMALL  "
90 '───────── MAP SET ─────────
100 A=1:B$=" ":C=1:FOR I=AD+FL*400 TO AD+FL*400+399:B$=B$+CHR$(PEEK(I))
110 C=C+1:IF C=21 THEN B$=B$+" ":M$(A)=B$:A=A+1:B$=" ":C=1
120 NEXT I:COLOR 7:IF FL<>2 THEN GOTO 260
130 IF F2>6 THEN M$(2)=MID$(M$(2),1,13)+"●●●●●●♣時 "
140 GOTO 260
150 '───────── PRINT STATES ─────────
160 LOCATE16,3:PRINTUSING "##";LV:LOCATE 15,5:PRINTUSING"###";HP
170 LOCATE15,7:PRINTUSING"###";MH:LOCATE 24,3:PRINTUSING"###";EX
180 LOCATE26,5:PRINTUSING  "#";FL:LOCATE 32,3:PRINTSW$
190 LOCATE32,5:PRINTAR$          :LOCATE 32,7:PRINTSH$
200 '───────── M O V E ─────────
210 S=STICK(ST):G=STRIG(ST):IF S+G=0 THEN 210
220 ON (G+2) GOTO 300
230 XX=0:YY=0:XX=XX-(S=6)+(S=4):YY=YY-40*(S=2)+40*(S=8)
240 C=PEEK@(&H30CC+XX+YY)
250 IF C<247 THEN MX=MX+XX:MY=MY+YY/40
260 '───────── PRINT MAP & ENEMY? ─────────
270 FOR I=0 TO 4:MA$(I)=MID$(M$(MY+I-3),MX-2,5):NEXT I
280 FOR I=0 TO 4:LOCATE 2,I+3:PRINT MA$(I):NEXT
290 ON INT(RND*20) GOSUB 390:GOTO 150
300 '───────── SEARCH FLOOR ─────────
310 SOUND 7,&B110110:PLAY"O2C1RCRC":PR$="ｱﾀﾘ ｦ ﾐﾃﾐﾀ..."
320 PR=0:GOSUB 350:C=ASC(CHARACTER$(4,5)):PAUSE 8
330 IF C=224 OR C=226 OR C=227 THEN PR$="ﾅﾆﾓ ﾐｱﾀﾗﾅｲ":GOSUB 350:GOTO 150
340 GOSUB CHR$(C)
350 '───────── PRINT MESSAGE ─────────
360 CONSOLE 11,13,1,24:CLS
370 FOR I=1 TO LEN(PR$):PRINT CHR$(ASC(MID$(PR$,I,1))-PR);:BEEP1:BEEP0
380 FOR T=0 TO 10:NEXT T,I:PRINT:CONSOLE:RETURN
390 '───────── FIGHT MODE ─────────
400 SOUND 7,&B111100:PLAY"O3E0CECECECE3":C$=CHR$(29,29,29,29,31)
410 ON FF GOTO 430,440:TE=INT(RND*LV+1):LOCATE 35,21:IF TE>9 THEN 410
420 PRINT CHR$(88+TE*4,89+TE*4,29,29,31,90+TE*4,91+TE*4):GOTO 450
430 TE=10:LOCATE 34,19:PRINT "▁▂▃";C$;"▃▅▆";C$;"╎┃▌▌";C$;"▌▌▌╱":GOTO 450
440 TE=11:LOCATE 34,19:PRINT "─┤┼┬";C$;"┤┼┼┐";C$;"┘└╭";C$;"◡ ╱╲"
450 PR$=MO$(TE)+"ｶﾞｱﾗﾜﾚﾀ!!":PR=0:GOSUB 350:TH=MH(TE):IF TE<>6 THEN 470
460 PAUSE 5:PR$="ｺﾞｰｺﾞｰﾗｯｶｾｲ ﾊ ｺﾞｰｺﾞｰ ｦ ｵﾄﾞｯﾃｲﾙ!!!":GOSUB 350
470 COLOR3:PRINT:PRINT" ﾃﾝｷｰ ﾃﾞ ﾀﾀｶｯﾃｸﾀﾞｻｲ!!":COLOR7
480 LOCATE 31,20:PRINT"  ";CHR$(29,29,31)"  ";CHR$(29,29,31);"  "
490 AP=0:SA=STICK(0):ON SA GOTO 520,510,500,520,510,500,520,510,500:GOTO 490
500 HI=SA/3:LOCATE 31,23-HI:PRINT"[":AP=INT(RND*SW*.4+SW*.6):KO=1:GOTO 530
510 HI=(SA+1)/3:LOCATE 31,23-HI:PRINT"〒":KO=2:GOTO 530
520 HI=(SA+2)/3:KO=3
530 DP=INT(RND*AR*.3+AR*.7):ON MT(TE) GOSUB 540,550:GOTO 620
540 HT=INT(RND*3)+1:KT=INT(RND*2)+1:RETURN
550 IF KO=3 THEN HT=HI:KT=1:GOTO 610
560 ON INT(RND*4+1) GOTO 570,580,590,600
570 KT=2:HT=HI:GOTO 610
580 KT=2:HT=HI:GOTO 610
590 KT=2:HT=HI:GOTO 610
600 KT=INT(RND*2)+1:HT=INT(RND*3)+1
610 RETURN
620 IF KT=1 THEN LOCATE 32,23-HT:PRINT"[":GOTO 640
630 LOCATE 32,23-HT:PRINT"〒"
640 IF KO<>3 THEN 670 ELSE PR=0:PR$="ｱﾅﾀ ﾊ ﾊｼﾘﾀﾞｼﾀ!!!":GOSUB 350:PAUSE 5
650 IF HI<>HT THEN PR$="ﾖｼ!! ﾆｹﾞﾚﾀｿﾞ!!":GOSUB 350:LINE(31,19)-(37,22)," ",BF:GOT
O 150
660 PR$="ﾀﾞﾒﾀﾞ... ﾂｶﾏｯﾀ!!":GOSUB 350:HP=HP-LV:GOTO 790
670 TA=INT(RND*MH(TE)*.2+MH(TE)*.2)
680 IF KO=1 AND KT=1 THEN HP=HP-TA:TH=TH-AP:PR$="ｳﾜｯ! ｱｲｳﾁﾀﾞ!!"        :GOTO 750
690 IF KO=2 AND KT=2 THEN HP=HP+SH*LV:PR$="ﾝ? ﾅﾝﾀﾞｱｲｺｶ.."            :GOTO 780
700 IF KO=1 AND KT=2 AND HI= HT THEN PR$="ｶﾜｻﾚﾀ!!"                 :GOTO 760
710 IF KO=1 AND KT=2 AND HI<>HT THEN TH=TH-AP:PR$="ｱﾀｯﾀ!!"          :GOTO 770
720 IF KO=2 AND KT=1 AND HI= HT THEN HP=HP+SH*LV:PR$="ｱﾌﾞﾈｰ ｳﾏｸﾖｹﾀｿﾞ!":GOTO 760
730 IF KO=2 AND KT=1 AND HI<>HT THEN HP=HP-TA:PR$="ｸﾞｴｯ ﾔﾗﾚﾀ..."      :GOTO 750
740 BEEP:END
750 PALET 0,2:PALET:SOUND 7,&B110000:PLAY"O4D0C:O2D0C":GOTO 780
760                 SOUND 7,&B110110:PLAY"O8D2:O7D1"  :GOTO 780
770                 SOUND 7,&B110000:PLAY"O4D0C:O2D0C"
780 GOSUB 350:IF HP>MH THEN HP=MH
790 LOCATE 15,5:PRINTUSING"###";HP
800 IF INKEY$="" AND HP>0 AND TH>0 THEN 800 ELSE IF HP>0 AND TH>0 THEN 480
810 IF HP<1 THEN 880
820 PR$="ﾔｯﾀ! "+MO$(TE)+"ｦﾀｵｼﾀ!!":GOSUB 350:LINE(31,19)-(37,22)," ",BF
830 ON FF GOTO 1300,1380
840 EX=EX+ME(TE):IF EX>999 THEN EX=999:GOTO150 ELSE IF EX<NL THEN150
850 PAUSE5:SOUND 7,&B111110:PLAY"O4C1DEDEDG:O4R0C1DEDEDG":EX=0
860 PR$="LEVEL ｶﾞ ｱｶﾞﾘﾏｼﾀ｡":GOSUB 350:LV=LV+1:NL=NL+30:MH=LV*10:HP=MH
870 GOTO 150
880 '───────── D E A D ─────────
890 FOR I=1 TO 5:PALET 0,2:PAUSE 1:PALET:PAUSE 1:NEXT:EX=INT(EX/2)
900 COLOR 5:PR=0:PR$="ｱﾅﾀ ﾊ ｼﾝﾃﾞｼﾏｯﾀ!!":GOSUB 350:COLOR 7:PAUSE 15
910 HP=MH:LINE(31,19)-(37,22)," ",BF:LOCATE 15,5
920 PRINTUSING"###";MP:MX=3:MY=3:FL=1:LINE(1,11)-(24,23)," ",BF:FF=0:GOTO 90
930 '───────── BONUS ─────────
940 LABEL "○":LABEL "◆":PR=0:PR$="ﾅﾝﾀﾞｶ ｹﾞﾝｷ ｶﾞ ﾃﾞﾃｷﾀ｡":GOSUB350:HP=MH:GOTO150
950 '───────── CLIMB & DOWN ─────────
960 LABEL "♣":IF F1>FL THENPR=0:PR$="ﾖｼ｡ ｳｴﾆﾉﾎﾞﾚﾙｿﾞ｡":GOSUB350:FL=FL+1:GOTO90
970 PR=0:PR$="ﾏﾀﾞ ｳｴﾆﾉﾎﾞﾚﾅｲ｡":GOSUB 350:GOTO 150
```

▶ただいま，X1turbo用シューティングゲーム「SUPERレイドット」を制作中。これは自機，敵キャラなどすべてが1ドットで描かれている，恐ろしく目に悪いゲームです。できたら，すぐに送ります。
中村　岳夫（16）東京都

```
980 LABEL "◢":PR=0:PR$="ｺﾉｱﾅｶﾗ ｼﾀ ﾉ FLOOR ﾆｲｹﾙﾅ｡":GOSUB 350:FL=FL-1:GOTO 90
990 '──────── MAN 1 ────────
1000 LABEL "◣":IF F2=0 OR F2=1 THEN PR=1:PR$=ME$(1):GOSUB 350:F2=1:GOTO 150
1010 IF F2=2 OR F2=3 THEN PR=1:PR$=ME$(3):GOSUB 350:F2=3:F3=1:GOTO 150
1020 IF F2=4 THEN PR=1:PR$=ME$(3):GOSUB 350:GOTO 150
1030 IF F2>=5 THEN PR=1:PR$=ME$(5):GOSUB 350:GOTO 150
1040 '──────── MAN 2 ────────
1050 LABEL "×":IF F2=1 OR F2=2 THEN PR=1:PR$=ME$(2):GOSUB 350:F2=2:GOTO 150
1060 IF F2=4 THEN PR=1:PR$=ME$(4):F2=5:GOSUB 350:LOCATE 27,12:PRINT"          "
:AR=AR+10:AR$="GREAT  ":F1=2:GOTO 150
1070 IF F2>=5 THEN PR=1:PR$=ME$(6):GOSUB 350:GOTO 150
1080 IF F2=0 OR F2=3 THEN PR=1:PR$=ME$(2):GOSUB 350:GOTO 150
1090 '──────── MAN 3 ────────
1100 LABEL "▘":IF F2<4 THEN PR=1:PR$=ME$(7):GOSUB 350:GOTO 150
1110 PR=1:PR$=ME$(8):GOSUB 350:GOTO 150
1120 '──────── MAN 4 ────────
1130 LABEL "▖":PR=1:PR$=ME$(9):GOSUB 350:GOTO 150
1140 '──────── MAN 5 ────────
1150 LABEL "▝":IF F2=5 THEN PR=1:PR$=ME$(10):GOSUB 350:F3=2:GOTO 150
1160 IF F2=6 THEN PR=1:PR$=ME$(11):GOSUB 350:F2=7:LOCATE 27,12:PRINT"
 ":SW=SW+10:SW$="LONG   ":F1=3:GOTO 90
1170 IF F2=>7 THEN PR=1:PR$=ME$(12):GOSUB 350:GOTO 150
1180 '──────── MAN 6 ────────
1190 LABEL "▗":PR=1:PR$=ME$(13):GOSUB 350:GOTO 150
1200 '──────── MAN 7 ────────
1210 LABEL "▞":IF F2=7 AND F3=3 THEN PR=1:PR$=ME$(15):GOSUB 350:GOTO 150
1220 IF F2=7 THEN PR=1:PR$=ME$(14):GOSUB 350:F3=3:GOTO 150
1230 IF F2=8 THEN PR=1:PR$=ME$(16):GOSUB 350:F2=9:LOCATE 27,12:PRINT"
 ":SH=SH+3:SH$="HYPER  ":F1=4:GOTO 150
1240 IF F2>=9 THEN PR=1:PR$=ME$(17):GOSUB 350:GOTO 150
1250 '──────── MAN 8 ────────
1260 LABEL "▚":IF F2<8 THEN PR=1:PR$=ME$(18):GOSUB 350:GOTO 150
1270 PR=1:PR$=ME$(19):GOSUB 350:GOTO 150
1280 '──────── MAN 9 ────────
1290 LABEL "□":IF F2=9 THEN FF=1:PR=1:PR$=ME$(20):GOSUB 350:PAUSE 80:GOTO 390:EL
SE C=224:GOTO 330
1300 PAUSE 10:FF=0:PR=1:PR$=ME$(21):GOSUB 350:HP=MH:F2=10:F1=5:F3=4:GOTO 150
1310 '──────── MAN 10 ────────
1320 LABEL "▒":IF F2<=10 THEN PR=1:PR$=ME$(22):GOSUB 350:GOTO 150
1330 IF F2=11 THEN PR=1:PR$=ME$(23):GOSUB 350:F3=5:F2=12:F1=5:GOTO 150
1340 IF F2=>12 THEN PR=1:PR$=ME$(24):GOSUB 350:GOTO 150
1350 '──────── FIRE ────────
1360 LABEL"±":IF F2<>13 THEN FO=1:GOSUB 1390:GOTO 150
1370 PR=1:PR$=ME$(25):GOSUB 350:FF=2:PAUSE 40:GOTO 390
1380 PR=0:PR$="ﾂ....ﾂﾖｲ.... (ｼﾝﾀﾞ)":GOSUB 350:GOTO 1680
1390 '──────── ITEM FOUND ? ────────
1400 ON FO GOSUB 1410,1420:RETURN
1410 PR=0:PR$="ﾅﾆｶ ｱﾘｿｳﾀﾞﾅ...":GOSUB 350:RETURN
1420 PR=0:PR$="ﾝ... "+FO$+" ｶﾞｵﾁﾃｲﾙｿﾞ..":GOSUB 350:LOCATE 27,12:PRINT FO$:SOUND
7,&B111100:PLAY"O4D1CBAB+C+D:O4R0D1CBAB+C+D":RETURN
1430 '──────── ITEM 1 ────────
1440 LABEL "金":IF F3<>1 THEN FO=1:GOSUB 1390:GOTO 150
1450 FO=2:FO$="ｼｮｳｷ ﾉ ｺﾅ":GOSUB 1390:F3=0:F2=4:GOTO 150
1460 '──────── ITEM 2 ────────
1470 LABEL "木":IF F3<>2 THEN FO=1:GOSUB 1390:GOTO 150
1480 FO=2:FO$="ｷﾝ ﾉ ｵﾉ":GOSUB 1390:F3=0:F2=6:GOTO 150
1490 '──────── ITEM 3 ────────
1500 LABEL "水":IF F3<>3 THEN FO=1:GOSUB 1390:GOTO 150
1510 FO=2:FO$="ﾖｳｶﾞﾝ ﾉ ｺﾅ":GOSUB 1390:PAUSE 15:F3=0:F2=8:PR=0
1520 PR$="ｺﾝﾅﾄｺﾆｵﾁﾃｲﾀﾉｶ...ﾐﾆｸｲﾅｧ":GOSUB 350:GOTO 150
1530 '──────── ITEM 4 ────────
1540 LABEL "火":IF F3<>4 THEN FO=1:GOSUB 1390:GOTO 150
1550 FO=2:FO$="ｷﾞﾝ ﾉ ﾂﾎﾞ":GOSUB 1390:F3=0:F2=11:GOTO 150
1560 '──────── ITEM 5 ────────
1570 LABEL "月":IF F3<>5 THEN FO=1:GOSUB 1390:GOTO 150
1580 PR=0:PR$="ICE SWORD ｦ ﾐﾂｹﾀｿﾞ!!!!":GOSUB 350:SW$="ICE SW ":SW=SW+20
1590 F2=13:GOTO 150
1600 '──────── SAVE & LOAD ────────
1610 LABEL "S":PR=0:PR$="SAVE OR LOAD             1:SAVE GAME               2:LOAD
GAME          3:EXIT":GOSUB 350
1620 A=STICK(0):IF A=0 THEN 1620 ELSE ON A GOSUB 1630,1650,1670:HP=MH:GOTO150
1630 OPEN "O",#1,"FLAME DATA":WRITE#1,F1,F2,F3,MH,LV,NL,SW,AR,SH,SW$,AR$,SH$
1640 CLOSE #1:RETURN
1650 OPEN "I",#1,"FLAME DATA":INPUT#1,F1,F2,F3,MH,LV,NL,SW,AR,SH,SW$,AR$,SH$
1660 CLOSE #1:EX=0:RETURN
1670 BEEP:RETURN
1680 '──────── ENDING ────────
1690 SOUND 7,&B111000:PLAY "O6C2-B-A7C2-B-A5-BCD7-A3E8:O6R0C2-B-A7C2-B-A5-BCD7-A
3E8":RESTORE 1790
1700 CLS 4:PRW &B11111110:C=0:FOR I=1 TO 7:PALET I,0:NEXT
1710 FOR I=0 TO 319:IF C=8 THEN C=0
1720 LINE(I,80)-(I,87),PSET,C:C=C+1:NEXT:GOTO 1750
1730 A=2 :FOR I=1 TO 6:B=256-A:A=A*2:PRW B:PAUSE1:NEXT:PRW 0:RETURN
1740 A=64:FOR I=1 TO 6:B=256-A:A=A/2:PRW B:PAUSE1:NEXT      :RETURN
1750 FOR J=1 TO 21:READ X,M$:LOCATE X,10:PRINT M$:GOSUB 1730:PAUSE (LEN(M$)*2)
1760 IF J<>21 GOSUB 1740:CLS:PAUSE 5:NEXT
1770 A$="O4C1DEDCEFFEDEFBAFE":B$="DEFGA4B4+C7":PLAY A$+A$+B$+":O4R0"+A$+A$+B$
1780 GOTO 1780
1790 DATA 9,｢ｾｲﾚｲ ﾉ ﾐｽﾞ｣ ｦ ｶｹﾙﾄ,5,｢ｼﾞｺﾞｸ ﾉ ﾄﾓｼﾋﾞ｣ ﾉ ﾎﾉｵ ｶﾞ ｷｴﾃﾕｸ｡
1800 DATA 7,ｷﾐ ﾊ ﾄｳ ﾉ ｵｸｼﾞｮｳ ﾆ ﾉﾎﾞｯﾀ｡,10,ﾑﾗ ﾉ ｻﾜｷﾞｺﾞｴ ｶﾞ ｷｺｴﾙ｡
1810 DATA 10,ﾓﾄ ﾉ ﾑﾗ ﾆ ﾓﾄﾞｯﾀﾝﾀﾞ｡,15,ﾄｳ ｦ ｵﾘﾖｳ｡
1820 DATA 14,ｿｳ ｵﾓｯﾀﾄｷ...,6,ﾑﾗ ﾆ ｳﾂｸｼｲ ﾎﾉｵ ｶﾞﾓｴﾃｲﾙﾉｶﾞﾜｶｯﾀ｡
1830 DATA 16,- STAFF -,17,PROGRAM,20,&,16,SCENARIO,13,MAKOTO TERAKAWA
1840 DATA 16,CHARACTER,20,&,15,TEST PLAY,15,HIDEKI UDA
1850 DATA 18,AND,18,YOU,9,THANK YOU FOR PLAYING,16,THE END
1860 '┌─────────────────────────────┐
1870 '│   1988 / 8 / 25     PM 9:30 │
1880 '│ SPECIAL THANKS ... HIDEKI UDA │
1890 '│     E          N         D  │
1900 '└─────────────────────────────┘
```

Z80マシン語ゲーム工房
最終回

爆発，そして完成へ

Murata Toshiyuki 村田 敏幸

約半年にわたって連載してきたこの「Z80マシン語ゲーム工房」も，いよいよ最終回を迎えます。今月は自機の当たり判定のほかにも細かい部分に改良を施し，ゲームとしての完成度をより高めていきます。さあ，あとひとふんばり。今回のこの講座をクリアした皆さんだけが，このゲームの全貌を目にすることができるのです。

前回まででゲームの基本部分はほぼ完成した。あとは自機が「ちゃんとやられる」ようにすれば出来上がりだ。最終回である今月は，まずこいつをやっつけて，おまけに得点の表示処理を付け加えたところで完成とし，それから，敵の種類を増やすなどのバージョンアップを試みる。

ところで，先月のリスト（注1）は読み終わったかな？

注1）先月のリストの中では，まだ話ていない命令をいくつか使っていた。命令表を見れば働きはわかると思うが，一応ここでフォローしておく。

CPLとNEGはそれぞれAレジスタの値の「1の補数」と「2の補数」を求め，Aレジスタに格納し直す命令だ。別の言い方をすると，CPLはAレジスタの全ビットを反転し，NEGは0からAレジスタの値を引く，つまり－Aを求める命令だ。CPLではゼロフラグが変化しないということを付け加えておく。

あと，「EX (SP), HL」ってのは，スタックからHLにポップすると同時にHLの値をPUSHする命令で，結局，スタックの一番上に積まれた2バイトと，HLの内容が交換されることになる。交換だから，もう一度繰り返せば元に戻る。

MZ-1500への対応

おっと，先にMZ-1500への対応をすませておく。いままでMZ-1500はMZ-700と共通のプログラムを使ってもらっていたが，MZ-1500にはPCGがあるわけだから，これを活用できるようにしてしまおう。

MZ-1500は40×25文字のテキスト画面を2画面分持ち（以下，画面1，2と呼ぶ），画面1は通常の文字用，2はPCG用に使われている。また，2つの画面は優先順位を付けて重ね合わせて表示することができる。が，画面2のみを表示するモードはないようだ。ゲームではPCGのみを使おうとしているのでこれはちょっと問題だが，通常の文字を使わないのだから，画面1の文字色と背景色を全部黒にすることでごまかすことができるだろう。

MZ-1500のPCGは，ほかに類を見ない強力さで，PCG 1～4それぞれ256個，計1024個が使え，当然それらは1文字あたり8色で表示を行うことができる。また，8色中8色のパレット機能も備えている。このあたりはPCGをテキスト画面にビッシリ並べて，グラフィック画面のように使うというMZ-1500の仕様からすると当然ともいえるだろう。ゲームではこのうちのPCG1の80_H～FF_Hを用いるようにする。

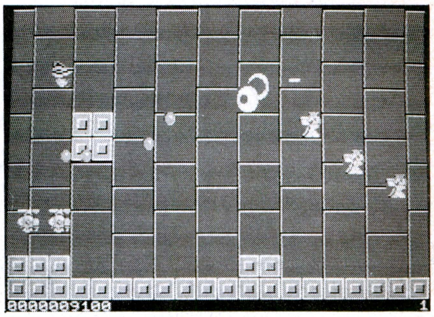
PCGの表示は，画面2のアトリビュートVRAMで制御される。第7，6ビットでPCGの1～4のどれを使うかを決め（00ならPCG1，01ならPCG2……），第3ビットで表示を行うかどうかを選択する（1なら表示)。残りのビットは0にしておくことになっているようだ。

ゲームではPCG1のみを使い，常に表示するわけだから，第3ビットだけが1であるようなデータ＝08_HでアトリビュートVRAMを埋めておけば，あとはテキストVRAMにPCGコードを書き込むだけです む。

PCGの定義は，CG-RAMにパターンデータを書き込むことで行う。CG-RAMは$D000_H$以降のバンクメモリ上に置かれており，このバンク切り換えはI/Oポート$E5_H$で行う。CG-RAMは青赤緑のプレーンごとに別バンクになっており，さらにこのアドレスにはCG-ROMも別バンクに存在する。たとえば，ポート$E5_H$に0を出力するとCG-ROMが，1を出力すると青プレーンのCG-ROMが……というようにバンク切り換えが行われる。バンクをCG-ROM/RAMから通常の状態に戻すには，I/Oポート$E6_H$になんでもいいからデータをOUTしてやればよい。

なお，CG-RAMの1バンクは先頭から8×256バイト単位でPCG 1～4に割り当てられており，また，それぞれは8バイトごとに区切られ，00_H～FF_Hのパターンを格納するようになっている。

パレットの設定はI/Oポートの$F1_H$で行う。ここに「パレットコード×16＋設定するカラーコード」をOUTすればパレットの設定が行われる。たとえば，71_Hを出力すれば，標準の状態で白の部分が青になるわけだ。



あと，表示画面の選択はI/Oポート$F0_H$を利用する。このポートに出力するデータの第0ビットは画面2を表示するかどうかを表し（1なら表示），第1ビットで重ね合わせの優先順位を設定する（0なら画面1優先，1なら画面2優先）。

といったところで実際のプログラムがリスト1だ。このリストと共通部，それからキャラクタパターン（これはX1などと共通），さらにMZ-700用のキー入力ルーチンをまとめてアセンブルしてもらいたい。共通部はまったく変更しないで使い，MZ-700専用の追加変更を加えてはならない点に注意してほしい。

リストは上で述べた知識があれば読みこなせると思うが，簡単に説明しておくと，INIT1500が画面の初期化ルーチン（アトリビュートの設定や画面の消去）で，DEF CHRが128個のPCGを定義するサブルーチン，また，DISPが仮想VRAMから実画面への一括転送ルーチンだ。画面2に対して書き込みを行うわけだから，VRAMの先頭アドレスは$D400_H$になる。

自機と敵との当たり判定

自機は，敵にぶつかったらその敵を道連れに爆発する。このうちの「爆発する」という部分はちょっと置いといて，とにかく自機と敵が重なったかどうかをチェックす

るサブルーチンを作っておこう。

どうやるかは，いまさら考えるまでもない。すでに自弾と敵との当たり判定処理のときやったように，お互いの座標を比べてみればいい。いま自機も敵機も2×2文字分の大きさを占めているから，敵のX，Y座標がどちらも自機の±1の範囲にあるときに両者は重なる。つまり，

1)　自機のX座標から敵のX座標を引き，結果が－1～1か？

2)　自機のY座標から敵のY座標を引き，結果が－1～1か？

という2段階のチェックで，ともにクロと出れば重なっていることがわかる。

ここで少々イヤラしいのが，減算結果が－1～1かどうかを調べる部分だ。ストレートにやると，

```
CP      2
JR      C, 範囲内
CP      0FFH
JR      NZ, 範囲外
```

範囲内：～

という感じになるところだろう(注2)。が，次のような方法もある。

```
INC     A
CP      3
JR      NC, 範囲外
```

範囲内：～

見てのとおり，比較の前に1を足しておくことで－1～1を0～2にずらしておき，

リスト1

```
;------------------------------------------------------------
;         機種依存部        MZ-1500用
;
;
          ORG     04E00H
;
START:    CALL    DEFCHR              ;PCG定義
          CALL    INIT1500                ;画面の初期化
          JP      MAIN                ;メインルーチンへ
;
;         画面の初期化
;
INIT1500:
          OUT     (0E3H),A            ;念のためVRAMへバンク切り換え
          LD      A,3                 ;画面2表示かつ優先
          OUT     (0F0H),A            ;
          LD      HL,0D400H           ;テキストVRAM第2画面を
          LD      DE,0D401H           ;スペースのキャラクタで
          LD      BC,0400H            ;埋める
          LD      (HL),SPCHR          ;
          LDIR                        ;
          LD      BC,0400H-1          ;アトリビュートVRAM第1画面を
          LD      (HL),00H            ;黒で埋める
          LDIR                        ;
          LD      BC,0400H-1          ;アトリビュートVRAM第2画面を
          LD      (HL),08H            ;PCG1表示で埋める
          LDIR
          RET
;
;         仮想VRAM->実VRAM転送
;          (VRAMはD000H以降)
;
DISP1500:
DISP:     LD      HL,4*48+4+VRAMMODOKI
                                      ;HL=仮想VRAM(4,4)のアドレス
          LD      DE,0D400H           ;D=実VRAM(0,0)アドレス
          EXX
          LD      B,24                ;画面縦サイズ
DISP0:    EXX
          LD      BC,40               ;実画面横サイズ
          LDIR                        ;横40文字分ブロック転送
          LD      C,8                 ;HL=HL+8
          ADD     HL,BC               ;
          EXX
          DJNZ    DISP0
          RET
;
;         PCG定義
;
DEFCHR:   LD      A,1                 ;PCG青選択
          OUT     (0E5H),A            ;
          LD      HL,PCGPAT           ;PCGパターン青
          CALL    SETCHR              ;1色分定義
          LD      A,2                 ;PCG赤選択
          OUT     (0E5H),A            ;
          LD      HL,PCGPAT+8         ;PCGパターン赤
          CALL    SETCHR              ;1色分定義
          LD      A,3                 ;PCG緑選択
          OUT     (0E5H),A            ;
          LD      HL,PCGPAT+16        ;
          CALL    SETCHR              ;1色分定義
          OUT     (0E6H),A            ;バンク標準
          RET
;
SETCHR:   LD      DE,0D400H           ;PCGコード80H先頭
          LD      B,128               ;128個のPCGを定義する
STCHR0:   PUSH    BC                  ;ループカウンタを待避する
          LD      BC,8                ;パターンサイズは8バイト
          LDIR                        ;ブロック転送
          LD      C,16                ;他2色のパターンを
          ADD     HL,BC               ;  スキップする
          POP     BC                  ;ループカウンタ復帰
          DJNZ    STCHR0              ;128回繰り返す
          RET
;
;         パレット初期化
;
INIPLT:   XOR     A                   ;パレットコード0
                                      ;カラーコード0
          LD      BC,0811H            ;B=ループカウンタ
                                      ;C=色の増分
INIPL0:   OUT     (0F1H),A            ;1色定義
          ADD     A,C                 ;パレットコードと
                                      ;  カラーコードを1増やす
          DJNZ    INIPL0              ;8色分繰り返す
          RET
```

リスト2

a

```
;
;         自機と敵との当たり判定
;
MSECHK:   LD      IY,CHRBUF           ;IY=敵バッファ先頭
          LD      DE,BUFSIZ           ;DE=敵バッファサイズ/1機
          LD      HL,(MSX)            ;L=自機X,H=自機Y
          LD      B,CHRMAX            ;B=敵の最大数
MSECK0:   LD      A,(IY)              ;A=敵タイプ
          INC     A                   ;死んでいる?
          JR      Z,MSECK3            ;そうならスキップ
          LD      A,L                 ;A=(MSX)
          SUB     (IY+X)              ;敵のX座標との
          INC     A                   ;  差が
          CP      3                   ;  -1～+1か?
          JR      NC,MSECK3           ;そうでなければ重ならない
          LD      A,H                 ;A=(MSY)
          SUB     (IY+Y)              ;敵のY座標との
          INC     A                   ;  差が
          CP      3                   ;  -1～+1か?
          JR      NC,MSECK3           ;そうでなければ重ならない
          ;
MSECK1:
          LD      (IY),DEAD           ;敵を殺す
          ;
MSECK2:   LD      A,1                 ;自機がやられたという
          LD      (MSFLAG),A          ;  フラグを立てる
          SCF                         ;CY=1ならやられた印
          RET
;
MSECK3:   ADD     IY,DE               ;ポインタを進め
          DJNZ    MSECK0              ;敵の数だけ繰り返す
          OR      A                   ;CY=0なら生き残り
          RET
```

b

```
;
;         自機と敵弾との当たり判定
;
MSMCHK:   LD      IY,EMSLBUF          ;IY=敵弾バッファ先頭
          LD      HL,(MSX)            ;L=自機X,H=自機Y
          LD      DE,EMBFSIZ          ;DE=バッファサイズ/1発
          LD      B,EMSLMAX           ;B=敵弾最大数
MSMCK0:   LD      A,(IY+EMX)          ;A=敵弾X座標
          CP      44                  ;X座標は44以上か？
          JR      NC,MSMCK1           ;そうなら空き領域
          SUB     L                   ;敵弾のX座標との
          INC     A                   ;  差は
          CP      2                   ;  -1～0？
          JR      NC,MSMCK1           ;そうでなければ当たらない
          LD      A,(IY+EMY)          ;A=敵弾Y座標
          CP      28                  ;Y座標は28以上か？
          JR      NC,MSMCK1           ;そうなら空き領域
          SUB     H                   ;敵弾のY座標との
          INC     A                   ;  差は
          CP      2                   ;  -1～0?
          JR      NC,MSMCK1           ;そうでなければ当たらない
          JR      MSECK1
MSMCK1:   ADD     IY,DE               ;ポインタを進め
          DJNZ    MSMCK0              ;敵弾の数だけ繰り返す
          OR      A                   ;CY=0なら生き残り
          RET
```

先月号のバグ退治

タコなバグがいくつか発見された。その1は，MZ-2500のPCG定義ルーチンがなにを勘違いしてか64個しかPCG定義していないというものだ。これは単純ミスなので，リスト3の54行目の64を128に直してもらいたい。あと，いまごろ気づいたのが，向きコードによる敵の移動処理ルーチンの1行抜け。Y座標に変位を加えておきながらワークを更新していないという間抜けなミスだ。このため蛇行タイプの当たり判定がうまくいっていなかったようだ。リスト5の566行目の直後に，

```
LD        (IX+Y),A
```

の1行を挿入してほしい。

それから，敵弾の移動ルーチンがなんとなく不安に思えるので，リスト5の774行と775行の間に，754〜755行と同じものをコピーして追加しておいてもらえば万全だと思う。

CPが1回ですむようにしているわけだ。

というところでリスト2aが自機と敵の当たり判定サブルーチンだ。すべての敵に対して前記の比較を行い，重なっていたら敵を殺し，同時に自機の状態を表すフラグを1にしている。ついでにリスト2bは自機と敵弾との当たり判定サブルーチンだ。やっていることは同じだよ。

また，どちらのサブルーチンも，あとで使うときのことを考えて，自機が生きて戻ったらノンキャリ，死んだらキャリを立てて戻るようにしてある。

注2）3行目は，

```
INC      A
```

でも構わない。

リスト3

a

```
;
;           自機と背景との当たり判定
;
MSWCHK: LD        HL,(MSADR)        ;自機の表示アドレス
        CALL      MSWCK0            ;上半分をチェック
        JR        C,MSECK2          ;CY=1なら激突した
        LD        DE,48-1           ;左下を指すように調整
        ADD       HL,DE             ;
        CALL      MSWCK0            ;下半分をチェック
        RET       NC                ;CY=0ならセーフ
        JR        MSECK2
;
MSWCK0: CALL      ISWALL            ;左側のチェック
        RET       C                 ;CY=1なら激突
        INC       HL                ;アドレスを進めて
        JP        ISWALL            ;右側のチェック
```

b

```
;
;           障害物のチェック（ＰＣＧ用）
;
ISWALL: LD        A,(HL)
        CP        090H
        CCF
        RET       NC
        CP        0A0H
        RET
```

c

```
;
;           障害物のチェック（キャラグラ用）
;
ISWALL: LD        A,(HL)
        PUSH      BC
        PUSH      HL
        LD        C,A
        LD        B,0
        LD        HL,WTBL
        ADD       HL,BC
        LD        A,(HL)
        POP       HL
        POP       BC
        ADD       A,A                           ;A= 0ならCY=0
                                                ;A=-1ならCY=1
        RET
;
WTBL:   DEFB      00,00,00,00,00,00,00,00 ;00
        DEFB      00,00,00,00,00,00,00,00
        DEFB      00,00,00,00,00,00,00,00 ;10
        DEFB      00,00,00,00,00,00,00,00
        DEFB      00,00,00,00,00,00,00,00 ;20
        DEFB      00,00,00,00,00,00,00,00

中　略

        DEFB      00,00,00,00,00,00,00,00 ;B0
        DEFB      00,00,00,00,00,00,00,00
        DEFB      00,00,00,00,00,00,00,00 ;C0
        DEFB      00,00,00,00,00,00,00,00
        DEFB      00,00,00,00,00,00,00,00 ;D0
        DEFB      00,00,00,-1,00,00,00,00             ;for 2000
        DEFB      00,00,00,00,00,00,00,00 ;E0
        DEFB      00,00,00,00,00,00,00,00
        DEFB      00,00,00,00,00,00,00,-1 ;F0         ;for 700
        DEFB      00,00,00,-1,00,-1,-1,00             ;
```

自機と背景との当たり判定

次に自機が壁とか床などの障害物に当たったかどうかを調べる処理を考える。今度は座標を比べるわけにはいかないというのは，すぐわかるだろう。キャラクタベースの単純なゲームを一度でも作った人ならわかるだろうが，こんなときは画面を直接読むのがてっとり早い。要するに自機を表示しようとしている仮想VRAMを調べて，そこに書いてある文字コードにより障害物かどうかを判断するわけだ。

リスト3aはこの方法により，自機と壁との当たり判定を行うサブルーチンだ。このリストはまだ未完成で，ISWALLというサブルーチン本体がない。このサブルーチンは「HLの指している仮想VRAMアドレスに書き込まれている文字が障害物であればキャリを立てて戻る」ものとしよう。あとは，どうやって障害物を構成する文字かどうかを判断するか，だ。

PCGを使う場合，あらかじめ何番から何番のPCGコードを障害物とみなすというように決めておけば，チェックが単純化できる。先月のリストの注釈に書いておいたように，PCGコード90_H〜$9F_H$を障害物と決めよう。すると，このチェックルーチンはリスト3bのようになる。

PCGを使わない場合はちょっと面倒だ。もちろんASCIIコードの何番〜何番のみを障害物に使うというような制限を付ければPCGと同じ方法が使えるのだが，なかなかそうもいかない。そこでリスト3cでは256バイトばかりのテーブルを用意することで切り抜けた。これは文字コード00_H〜FF_Hの順に，その文字が障害物のキャラクタを構成するものなら−1，そうでなければ0，というデータを並べたものだ。こうすれば簡単なテーブル参照の技法で，その文字コードが障害物か否かを調べられる。

なお，MZ-700の場合は仮想VRAMの文字コードを調べる代わりに，仮想アトリビュートVRAMを調べ，たとえば表示色が赤であれば障害物というように決める方法もある。キャラクタパターンデータの作り方次第ではどうにでもなるということだ。

以上，当たり処理ルーチンの組み込みは，ちょい先のリスト5でやっている。ここで，自機の当たり判定は本来なら自機が移動する前とあとの2回行うべきなのだが，プログラム中では自機周りの当たり判定は多少ルーズなほうがよいと思って，移動後の判定は勝手に省略させてもらった。

爆発させてみる

前回までのプログラムでは，敵はやられるとパッと消えるだけだった。やっぱり派手に爆発させてあげなきゃかわいそうだから，爆発の処理を作ってやろう。これはそう難しいことではない。敵の種類はバッファ先頭の1バイトで区別していた（0なら直進タイプ，1なら蛇行タイプというように）が，ここに「爆発というキャラクタのタイプ」を付け加えてしまえばよい。つまり，敵キャラはやられることによって「爆発というキャラクタに変身する」わけだ。

リスト4が爆発キャラクタの移動・表示処理だ。もっとも移動はせずに表示のみを行っているだけだが，もっと凝りたければ，徐々に落下させるとか，慣性に従って流れるように墜落させることもできるだろう。

爆発のキャラクタタイプコードには，30_H～$3C_H$を充てた。幅があるのは3段階の単純なアニメ処理を施したからだ。まだ数が合わないが，このカラクリはリストを見てもらいながら解き明かすことにしよう。

ラベルBOMB1がキャラクタタイプコード30_Hの処理ルーチンだ。HLにBMPAT1を入れて，BOMB4に分岐している。BMPAT1以下には爆発のキャラクタパターンが格納されていることを仮定している。同様にBOMB2がタイプ34_H，BOMB3が38_Hの処理だ。やはりキャラクタパターンの先頭アドレスをHLに入れてBOMB4で合流する。また，キャラクタタイプ$3C_H$を除く残りの場合はすべて，BOMB5に飛び込ませるようにする。

BOMB4では，まずHLに入っているキャラクタパターンアドレスをバッファ内に格納する。これにより，次にサブルーチンPUTを呼び出したときには，この新しくセットしたキャラクタパターンが表示されることになる。BOMB5ではキャラクタタイプをインクリメントしている。タイプ30_Hは31_Hに，31_Hは32_Hにそれぞれ「変身」するわけだ。それから仮想VRAMへの書き込みルーチンに分岐して爆発の処理は終わる。もうなにをやっているのかは見当がついたことだろうが，念のために爆発の過程を順に考えてみよう。

敵キャラはやられるとタイプ30_Hの爆発に変身する。タイプ30_Hの処理は先で述べたように，爆発のキャラクタデータを自分のバッファ内にセットし，タイプ31_Hに変身したうえで，表示を行うというものだ。タイプ31_Hに変身したので，次の回には直接BOMB4から処理を始める。やることはタイプ32_Hに変身してからパターンの表示を行うだけだ。もちろん表示されるパターンはタイプ30_Hのときにセットした爆発のパターンだ。こうやって変身を続けながら，やがてタイプ34_Hになる。今度はBOMB2から処理が始まるので，新しくBMPAT2を自分のキャラクタパターンとし，変身と表示を行う。

というように，メインループ4回に1回ごとにキャラクタパターンを3段階に切り換えながらキャラクタタイプはいつしか$3C_H$になる。タイプ$3C_H$の処理ルーチンはBOMB0だ。ここでは見てのとおり，キャラクタタイプを－1にする，つまり，本当に殺してしまう処理をしている。この時点でそのキャラクタデータバッファは空き状態に戻ることになる。

敵が爆発するようになったことで，ほかのサブルーチンも多少手直ししなければな

リスト4

```
;
;            敵キャラの爆発処理
;
BOMB1:  LD      HL,BMPAT1           ;爆発パターン1
        JR      BOMB4
BOMB2:  LD      HL,BMPAT2           ;爆発パターン2
        JR      BOMB4
BOMB3:  LD      HL,BMPAT3           ;爆発パターン3
BOMB4:  LD      (IX+PATL),L         ;新しいパターンを
        LD      (IX+PATH),H         ;  セット
BOMB5:  INC     (IX+TYPE)           ;タイプをインクリメント
        JP      PUT                 ;表示
;
BOMB0:  LD      (IX+TYPE),DEAD      ;本当に殺す
        RET
```

リスト5

```
;
;            自機の移動処理
;
MOVEMS:  LD      IX,MSBUF            ;IX=自機データバッファ先頭
         LD      A,(IX)
         OR      A
         JR      Z,MSMOV
         DEC     A
         ADD     A,A
         LD      E,A
         LD      D,0
         LD      HL,MSJTBL
         ADD     HL,DE
         LD      E,(HL)
         INC     HL
         LD      D,(HL)
         EX      DE,HL
         JP      (HL)
;
MSJTBL:  DEFW    MSBOM1,MSBOM5,MSBOM5,MSBOM5
         DEFW    MSBOM5,MSBOM5,MSBOM2,MSBOM5
         DEFW    MSBOM5,MSBOM5,MSBOM5,MSBOM5
         DEFW    MSBOM3,MSBOM5,MSBOM5,MSBOM5
         DEFW    MSBOM5,MSBOM5,MSREV
;
MSBOM1:  LD      HL,BMPAT1           ;爆発パターン1
         JR      MSBOM4
MSBOM2:  LD      HL,BMPAT2           ;爆発パターン2
         JR      MSBOM4
MSBOM3:  LD      HL,BMPAT3           ;爆発パターン3
         ;
MSBOM4:  LD      (MSPAT),HL          ;新しいパターンをセット
MSBOM5:  INC     (IX)                ;フラグをインクリメント
         RET
;
;            自機がやられた時の復活処理
;
MSREV:   LD      HL,LEFT             ;残機を
         DEC     (HL)                ;   1機減らす
         LD      A,(HL)              ;それをAに取り出して
         INC     A                   ;-1か?
         JP      Z,GAMEOVER          ;そうならゲームオーバー
         CALL    PRLEFT              ;残機を表示
         CALL    INITMS              ;自機のデータバッファを初期化
         ;
MSMOV:   LD      A,(MSWK3)           ;未使用だったMSWK3を
                                     ;   復活時の無敵状態カウンタに
         CP      40H                 ;40H未満?
         JR      NC,MSMOV0           ;そうでなければ無敵じゃない
         INC     A                   ;カウンタを進め
         LD      (MSWK3),A           ;   更新する
         JR      MSMOV1              ;当たり判定をスキップする
MSMOV0:  CALL    MSECHK              ;自機と敵との当たり判定
         RET     C                   ;CY=1ならやられた
         CALL    MSMCHK              ;自機と敵弾の当たり判定
         RET     C                   ;CY=1ならやられた
         CALL    MSWCHK              ;自機と壁の当たり判定
         RET     C                   ;CY=1ならやられた
MSMOV1:  CALL    INKEY               ;キー入力データを得る
         LD      DE,(MSX)            ;E=X座標，D=Y座標

中 略

         LD      (HL),MMSLCNT        ;リピートカウンタをリセット
FREND:   RET
TRGOFF:  LD      HL,MSWK1            ;今回、弾を撃たなかったという
         LD      (HL),0              ;        印をセット
MSMV9:   RET
;
;            自機の仮想VRAMへの表示
;
PUTMS:   LD      A,(MSWK3)           ;無敵状態カウンタの
         AND     1                   ;第0ビットが1か?
         RET     NZ                  ;そうなら表示しない（点滅用）
         LD      IX,MSBUF            ;IX=自機データバッファ先頭
         JP      PUT                 ;表示ルーチンへ
```

らなくなった。まず，今月のリスト2aの自機と敵との当たり判定では，敵キャラをいきなり殺さずに爆発に変身させるようにしなければならない。23行目のDEADを30_Hにしよう。それから9行と10行を次のように変更する。

```
        CP      30H
        JR      NC,～
```

これは自機と爆発との当たり判定をしないようにするための処置だ。

同様の変更を先月の自弾と敵との当たり判定サブルーチンにも施してもらいたい。手を加えるのは先月のリスト5の253，254，265行だ。

ゲームオーバーもある

自機もやっぱりやられると爆発する。んでもって，残機が1機減り，再びスタートする。もちろん，自機が全滅したらゲームオーバーだ。

爆発の処理は敵のときの方法がそのまま使える。残機を減らすのはワークエリアをデクリメントすればいい。このとき，画面に残機の数を表示し直すことにしよう。再スタートは，面倒だから突然元の位置に復活することにしてしまえ。復活してしばらくは無敵状態にするのが礼儀というものだろう。これには点滅させるのが一般的だな。

全滅したときは，なにも考えずにメインループを抜けてしまえばすむ。といいながら，どうやってメインループを抜けるかという問題は残る。ありきたりなのは全滅したというフラグを設けて，フラグが立っていたら抜ける，という方法だが，今回はあえて自機の移動ルーチンから，メインルーチンに戻ることなくいきなり終わらせてしまうという強行手段に訴えることにした。

自機の移動ルーチンはメインルーチンからサブルーチンコールされる。ここで普通にRETすれば，メインルーチンに戻るわけだ。ところが，ここでピョンとゲームオーバー処理ルーチンにジャンプしてしまうとする。ここからメインルーチンをすっ飛ばして，呼び出し元にリターンしてしまおうというわけだ。

ゲームオーバー処理ルーチンへ分岐した時点では，スタックの一番上には「自機の移動ルーチンからメインループへの戻りアドレス」が積まれている。そして，その1段下に「メインルーチンから，それを呼び出した親ルーチンへの戻りアドレス」が積まれていることになる。いま，メインルーチンの親へ戻りたいのだから，スタックの一番上のアドレスはもう必要がない。というわけで，適当なレジスタペアにこいつをポップしてしまおう。すると，スタックの一番上には親ルーチンへの戻りアドレスが残るから，RETすることで親ルーチンへ戻ることができる。

このようにスタックを調節することで，サブルーチンのサブルーチンから一気に親ルーチンまで戻るという方法は知っていて損はない。当然，サブルーチンの呼び出しが入れ子になっているときは，その数だけPOPを繰り返してつじつまを合わせればいい。

では，サブルーチンコール以外にもPUSHが多用されていて，スタックが山積みになっている場合はどうだろう。もちろん，スタックにどれだけの数のデータが積まれているか数えて，その分ダミーのPOPを並べてもよいわけだが，もうちょっとスマートな方法もある。リストAに簡単な例を示そう。

MAINでは，最初に現在のスタックポインタの値をワークエリアに格納し，サブルーチンSUB1を呼び出す。そして，SUB1はさらにサブルーチンSUB2を呼び出す。ここから普通にRETすれば，SUB1に戻り，さらにRETするとMAINに戻るはずだ。が，SUB2では先ほどしまっておいた，スタックポインタの値をスタックに代入してからリターンしている。これにより，SUB1もMAINも飛ばして，呼び出し元に返ることができるというわけだ。

といったところで，リスト5は自機の爆発，再スタートの処理ルーチンだ。上で書いたとおりの処理をしているのがわかるだろう。再スタート時に自機を点滅させるために自機の表示ルーチンにも手が加えられている。ちなみに，点滅させるのには2回に1回しか表示しなければよいのは当たり前。

なお，スタックのつじつま合わせを含むゲームオーバー処理ルーチンは，のちのリスト8のほうにまとめてある。また，PRLEFTは残機の表示を行うサブルーチンだが，次の得点表示ルーチンのところで一緒に作るので，ここのリストには含まれていない。

得点表示を考える

得点表示処理ルーチンを作る前に，一般的な10進数の表示の仕方をちょっとやっておく。1桁なら簡単だろう。ASCIIコードにしろMZ-700のディスプレイコードにしろ，数字には0～9の順番どおりに文字コードが与えられている。仮にAレジスタに0～9の値が入っているとすると，それに「0の文字コード」を足せば，「1桁の10進数を表す文字」に変換できる。ASCIIコードなら‘0’～‘9’には30_H～39_Hが割り当てられているから，30_Hを足せばよい。30_HとORを取っても同じことだ。

リストA

```
        ORG     8000H
;
MAIN:   LD      (SPWORK),SP
        CALL    SUB1
        RET
;
SUB1:   CALL    SUB2
        RET
;
SUB2:   LD      SP,(SPWORK)
        RET
;
SPWORK: DEFS    2
```

次に，2バイト数値を10進表示することを考えてみよう。2バイトでは0～65535までの最大5桁の数が表現できるわけだから，まず10000で割って万の位がいくつか調べ，それを1桁のときの方法で表示し，続いて1000で割って……というような処理を繰り返せばよいだろう。逆に，10で割ってその余りをどこかに格納しておき，さらに10で割って……という処理ののち，逆順で表示しても構わない。どちらの場合も頭に余分な0が付くが，手を加えさえすれば，最初の0ではない桁以降を表示するようにもできる。

さて，1バイトでは00_H～FF_Hの値が表現できるわけだ。16進と10進の相互変換ぐらいそらでできる読者も多いと思う。12_Hは10進では18で，29_Hに1を足せば$2A_H$だ，なんてことは書くだけ無駄というものだ。が，世の中には2進化10進数（BCD）というものがあって，この世界では12_Hを10進に直すと12になり，29_Hに1を足すと30_Hになるという。つまり，1桁の16進数では本来0～Fまでの数を表せるが，あえてA～Fを無視して見かけ上10進数のようなフリをしているわけだ。

このBCDによって表された数を10進表示するには，上の桁から順に4ビット（16進1桁）取り出して0の文字コードを足し表示，という処理を繰り返せばよい。割り算がない分，プログラムも簡単になるだろう。この利点を生かして，制作中のゲームでは得点はBCD表現でワークに格納することにした。

こうなると，BCD数同士の演算についても触れておかなければならないだろう。CPUによってはBCDデータ専用の演算命令が用意されているものもあるが，Z80に

はない。Z80ではBCD数の演算にも，ADDやSUBをそのまま使う。そして，加減算のあとに結果をBCDに補正するDAA（Decimal Adjust Accumulator）という命令を置く。たとえば，AレジスタにBCDで15（$=15_H$）が入っているときに，BCDの6をADDで足すと，結果は$1B_H$になってしまう。ここでDAAを実行すると，Aレジスタの値をポンと21_Hに直してくれるわけだ（注3）。

さらに多桁のBCD数の加算の仕方を考えておこう。ゲームのスコアが0〜99までというわけにはいかないからね。例として4桁（2バイト）のBCDに2桁（1バイト）のBCDを足す場合を取り上げる。

以前，「HL＋A」の計算でもやったように，こういうときはまず下位バイトを足し，続いてその桁上がり（＝キャリ）を上位バイトに足し込んでやるという手を使う。いま，HLとAにそれぞれ4桁，2桁のBCD数が入っているとすると，

```
        ADD     A,L
        DAA
        LD      L,A
        LD      A,H
        ADC     A,0
        DAA
        LD      H,A
```

リスト6

```
;.............................................................
;
;       得点・残機表示処理
;
ZEROCHR EQU     020H            ;'0'の文字コード(700)
;       EQU     030H            ;'0'の文字コード(2000)
;       EQU     0F0H            ;'0'の文字コード(2500,1500,X1)
SCOADR  EQU     40*24+0D000H    ;得点を表示するアドレス(2000/700)
;       EQU     40*24+0E000H    ;得点を表示するアドレス(2500)
;       EQU     40*24+0D400H    ;得点を表示するアドレス(1500)
;       EQU     40*24+3000H     ;得点を表示するアドレス(X1)
LFTADR  EQU     SCOADR+39       ;残機を表示するアドレス
;
;       得点の初期化
;
INITSCO:
        LD      HL,0            ;8桁とも0
        LD      (SCORE),HL      ;
        LD      (SCORE+2),HL    ;
        LD      A,ZEROCHR       ;A='0'
;       ..................X1シリーズでは以下を削除する
        LD      HL,SCOADR       ;得点表示アドレス
        ;..................2500では以下を復活する
        ;LD     DE,SCOADR+400H  ;得点表示アドレス（裏画面）
        ;..................2500ではここまでを復活する
        LD      B,10            ;10個'0'を並べる
INISC0: LD      (HL),A          ;ひたすら
        INC     HL              ;  書く
        ;..................2500では以下を復活する
        ;LD     (DE),A          ;裏画面にも
        ;INC    DE              ;  書く
        ;..................2500ではここまでを復活する
        DJNZ    INISC0
;       ..................X1シリーズではここまでを削除する
;       ..................X1シリーズでは以下を復活する
;       LD      BC,SCOADR       ;得点表示アドレス
;       EXX
;       LD      BC,SCOADR+400H  ;得点表示アドレス（裏画面）
;       LD      E,10            ;以下、
;INISC0:OUT     (C),A           ;  やっていることは
;       INC     BC              ;  MZと変わらない
;       EXX                     ;
;       OUT     (C),A           ;
;       INC     BC              ;
;       EXX                     ;
;       DEC     E               ;
;       JR      NZ,INISC0       ;
;       ..................X1シリーズではここまでを復活する
;
;       残機の初期化
;
INILFT: LD      A,3             ;残機を3に
        LD      (LEFT),A        ;  初期化
;
;       残機の表示
;
PRLEFT: LD      A,(LEFT)        ;残機は
        CP      10              ;  10以上か?
        JR      NC,PRLFT0       ;そうなら分岐
        ADD     A,ZEROCHR       ;'0'を足す
        JR      PRLFT1
PRLFT0: LD      A,ZEROCHR+9     ;10機以上は'9'を表示
PRLFT1:
;       ..................X1シリーズでは以下を削除する
        LD      HL,LFTADR       ;残機表示アドレスへ
        LD      (HL),A          ;  書き込み
        ;..................2500では以下を復活する
        ;LD     HL,LFTADR+400H  ;残機表示アドレス（裏画面）へ
        ;LD     (HL),A          ;  書き込み
        ;..................2500ではここまでを復活する
        RET
;       ..................X1シリーズではここまでを削除する
;       ..................X1シリーズでは以下を復活する
;       LD      BC,LFTADR       ;残機表示アドレスへ
;       OUT     (C),A           ;  書き込み
;       LD      BC,LFTADR+400H  ;残機表示アドレス（裏画面）へ
;       OUT     (C),A           ;  書き込み
;       RET
;       ..................X1シリーズではここまでを復活する
;
;       得点の加算・表示
;
SCOSUB: PUSH    BC
        PUSH    DE
        PUSH    HL
        CALL    GETSCO          ;得て
        CALL    ADDSCO          ;足して
        CALL    PRTSCO          ;表示する
        POP     HL
        POP     DE
        POP     BC
        RET
;
;       敵タイプに応じた点数を得る
;
GETSCO: LD      E,(IY+TYPE)     ;DE=敵タイプ
        LD      D,0             ;
        LD      HL,SCOTBL       ;HL=点数テーブル
        ADD     HL,DE           ;
        LD      A,(HL)          ;A=点数
        RET
;
;       得点の加算（BCD8桁）
;
ADDSCO: LD      HL,SCORE+3      ;最下位2桁
        LD      B,4             ;4バイト
        OR      A               ;CY=0
ADSCO0: ADC     A,(HL)          ;足して
        DAA                     ;10進補正して
        LD      (HL),A          ;更新
        DEC     HL              ;上位桁へ
        LD      A,D             ;D=0
        DJNZ    ADSCO0
        RET
;
;       得点の表示
;
;       ..................X1シリーズでは以下を削除する
PRTSCO: LD      A,ZEROCHR       ;A='0'
        LD      HL,SCORE        ;HL=得点最上位桁
        LD      DE,SCOADR
        LD      B,4
        ;..................2500では以下を復活する
        CALL    PRSCO0          ;表画面へ表示
        LD      HL,SCORE        ;同様の処理を
        LD      DE,SCOADR+400H  ;  裏画面に対しても
        LD      B,4             ;  行う
        ;..................2500ではここまでを復活する
PRSCO0: RLD                     ;上位4ビットを取り出し
        LD      (DE),A          ;表示
        INC     DE
        RLD                     ;下位4ビットを取り出し
        LD      (DE),A          ;表示
        INC     DE
        RLD                     ;元に戻すため
        INC     HL              ;下位桁へ
        DJNZ    PRSCO0
        RET
;       ..................X1シリーズではここまでを削除する
;       ..................X1シリーズでは以下を復活する
;PRTSCO:LD      A,ZEROCHR
;       LD      HL,SCORE
;       LD      DE,SCOADR
;       LD      B,4
;       LD      BC,SCOADR
;       EXX
;       LD      BC,SCOADR+400H
;       EXX
;       LD      E,4
;PRSCO0:RLD
;       CALL    PRSCO1
;       RLD
;       CALL    PRSCO1
;       RLD
;       INC     HL
;       DEC     E
;       JR      NC,PRSCO0
;       RET
;PRSCO1:OUT     (C),A
;       INC     BC
;       EXX
;       OUT     (C),A
;       INC     BC
;       EXX
;       RET
;       ..................X1シリーズではここまでを復活する
;
SCOTBL: DEFB    2,5,1,3,0,0,0,0 ;00-07
        DEFB    0,0,0,0,0,0,0,0 ;08-0F
        DEFB    0,0,0,0,0,0,0,0 ;10-17
        DEFB    0,0,0,0,0,0,0,0 ;18-1F
        DEFB    0,0,0,0,0,0,0,0 ;20-27
        DEFB    0,0,0,0,0,0,0,0 ;28-2F
;
SCORE:  DEFS    4
LEFT:   DEFS    1
```

によって，加算結果がちゃんとBCDでHLに入る。桁数が増えても同じような処理の繰り返しで正しい結果を得ることができる。

リスト６はゲームで使う得点（と残機）関係のサブルーチンだ。機種ごとに少しずつ違う部分があるのを無理に１本のリストにまとめてしまったため，ごちゃごちゃになってしまったが，注釈をよく見て自分の機種に合わせて変更を加えてもらいたい。敵のタイプに応じてテーブルから点数をAレジスタに取り出すGETSCO，その点数を得点に加えるADDSCO，さらに得点表示を行うPRTSCOからなる。

敵のタイプに応じた点数を得る部分については問題ないだろう。SCOTBL以下に用意された点数のテーブルから，配列参照の技法でもって，１バイト取り出してくるだけだ。現在の得点は，SCOREというラベルで示される行の４バイトのワークエリアに上位桁〜下位桁の順番で並んでいる。これにより，得点は最大10進８桁となる。

得点は画面の左下に表示することにした。画面の最下段は仮想VRAMの範囲外にしてあるので，この表示は直接実VRAMに書き込むことで行っている。画面上の(0, 24)から表示を始めるわけだから，そのVRAM上のアドレスは，

VRAMの先頭アドレス＋24×40

で求められる。８行のラベルでこのアドレスを定義しているが，この値は機種によって違うので注釈に示すように変更すること。また，その上のラベルZEROCHRは‘0’の文字コードを表している。これも機種によって異なるから，やはり注釈のように直す。なお，PCGを使う機種ではPCGの$F0_H$〜$F9_H$を数字に充てることにした。このPCGパターンはリストには含まれていないので，各自用意すること。

表示処理自体も特に変わったことをしているわけではない。得点を格納した４バイトのワークの頭から順に４ビットずつ取り出して，‘0’の文字コードと足してVRAMに書き込んでいる。ここで使っているRLDという命令は，「Aレジスタの下位４ビットとHLレジスタの指すメモリの１バイトの計12ビットを４ビット単位で左にローテートする」という，ちょっと変わった命令だ。右に回転するRRDとともにあまり使わない命令だけど，BCDを扱うときには知っていると便利なことが多い。

また，MZ-2500とX1では，画面０と１を交互に表示する関係で，両画面に得点を表示する必要があるのを忘れちゃいけない。

これらの処理は自機と敵，および，自弾と敵の当たり判定ルーチンから呼び出されることになる。具体的な組み込み方は次の節でまとめて話すことにする。

ところで，得点の表示は敵がやられたときにのみ行うわけだから，ゲームがスタートした直後にはなにも表示されていないことになる。それではみっともないから，リスト６には最初に得点表示位置に０を並べるサブルーチンINISCOを用意してある。得点は８桁までのはずなのに，このサブルーチンでは０を10個並べて表示している。後ろに余分な０を２つ表示することで，点数の単位が100点に見えるようにしているわけだ。高得点にこだわる人は，敵１機10万点とかにするのもお好み次第だ。

注３）　少し余計な話をしておくと，Z80にはN（サブトラクト）フラグというものがあり，直前に実行した演算が加算（ADD，ADC，INC）か減算

リスト７

```
JMPTBL: DEFW    TEKIA,TEKIB,DMY,  DMY    ;00-03
        DEFW    DMY  ,DMY,  DMY,  DMY    ;04-07
        DEFW    DMY  ,DMY,  DMY,  DMY    ;08-0B
        DEFW    DMY  ,DMY,  DMY,  DMY    ;0C-0F
        DEFW    DMY  ,DMY,  DMY,  DMY    ;10-13
        DEFW    DMY  ,DMY,  DMY,  DMY    ;14-17
        DEFW    DMY  ,DMY,  DMY,  DMY    ;18-1B
        DEFW    DMY  ,DMY,  DMY,  DMY    ;1C-1F
        DEFW    DMY  ,DMY,  DMY,  DMY    ;20-23
        DEFW    DMY  ,DMY,  DMY,  DMY    ;24-27
        DEFW    DMY  ,DMY,  DMY,  DMY    ;28-2B
        DEFW    DMY  ,DMY,  DMY,  DMY    ;2C-2F
        DEFW    BOMB1,BOMB5,BOMB5,BOMB5  ;30-33
        DEFW    BOMB2,BOMB5,BOMB5,BOMB5  ;34-37
        DEFW    BOMB3,BOMB5,BOMB5,BOMB5  ;38-3B
        DEFW    BOMB0                    ;3C
```

リスト８

```
;........................................................................
;
;       メインルーチン
;
MAIN:   LD      (SPWORK),SP     ;現在のスタックポインタを
                                ;  待避しておく
MAIN0:  LD      HL,IVNTDAT      ;イベントデータポインタの
        LD      (IVNTP),HL      ;      初期化
        LD      HL,EMSLCNT      ;敵弾移動カウンタの
        LD      (HL),0AAH       ;      初期化
        LD      A,88H
        LD      (IVNTC),A       ;イベントカウンタの初期化
        LD      (MAPCNT1),A     ;マップカウンタ1の初期化
        LD      (MAPCNT2),A     ;マップカウンタ2の初期化
        LD      HL,MAPDAT       ;マップデータポインタの初期化
        LD      (MAPPNT),HL     ;
        LD      HL,4*48+4+VRAMMODOKI
        LD      (HOME),HL       ;仮想VRAM上のHOME位置初期化
        CALL    INITMS          ;自機ワークの初期化
        CALL    INITMMSL        ;自弾ワークの初期化
        CALL    INITWK          ;敵ワークの初期化
        CALL    INITEMSL        ;敵弾ワークの初期化
        CALL    INITSCO         ;スコアの初期化
        CALL    EXMAP0          ;マップの展開と
        CALL    DISP            ;  表示（タイトル代わり）
STLOOP: CALL    INKEY           ;トリガー１が
        AND     020H            ;  押されるまで
        JR      NZ,STLOOP       ;  待つ
LOOP:   CALL    IVNT            ;イベント処理
        RET     C               ;CY=1であれば全面クリア
        CALL    EXTMAP          ;背景を仮想VRAMに展開
        CALL    MOVEMS          ;自機の移動
        CALL    MOVEMMSL        ;自弾の移動・表示
        CALL    MOVETEKI        ;敵の移動・表示
        CALL    MOVEEMSL        ;敵弾の移動・表示
        CALL    PUTMS           ;自機の表示
        CALL    DISP            ;仮想VRAMを表示
        CALL    PAUSE           ;一時停止
        LD      HL,3000         ;時間稼ぎの
WAIT:   DEC     HL              ;  空ループ
        LD      A,H             ;
        OR      L               ;
        JR      NZ,WAIT         ;
        JR      LOOP
;
;       ゲームオーバー処理
;
GAMEOVER:
        LD      SP,(SPWORK)     ;スタックを元に戻す
EDLOOP: CALL    INKEY           ;トリガー１が
        AND     020H            ;  離されるのを
        JR      Z,EDLOOP        ;  待つ
        RET                     ;ゲームを抜ける
;       JP      MAIN0           ;上のRETの代わりに復活させると
                                ;  呼び出したシステムに戻らずに
                                ;  ゲームを再スタートする
                                ;（ゲームを抜けるときはリセット）
;
SPWORK: DEFS    2               ;スタックポインタ待避用ワーク
```

(SUB, SBC, DEC, NEG) かを保持している。DAAはこれを参照し，15_H＋06_Hの直後であれば$1B_H$を21_Hに，20_H－09_Hの直後であれば$1B_H$を11_Hに，というように正しく補正する。なお，Nフラグはあくまで CPU内部で使われるフラグだから，これを分岐の条件とするような命令はない。

また，DAAはこのほかに下位4ビットからの繰り上がりがあったかどうかを示すH（ハーフキャリ）フラグ，および，お馴染み8ビット全体からの繰り上がりがあったかどうかを示すキャリフラグも調べて，補正を行っている。

一応の完成版

では今月作ったサブルーチンを本体に組み込もう。まず新規のサブルーチンをソースの適当な位置に挿入する。まさか「適当な位置とはどこだ」と聞く人はいないだろう。サブルーチンはどんな順序で並んでいても構わないわけだから，既存のサブルーチンの間ならどこでも構わない。

リスト3のISWALLは，MZ-700とMZ-2000ではcを，それ以外の機種はbを使う。また，リスト3cのテーブル部分は先月用意した背景パターンをそのまま使うことを前提にしているので，すでに背景を増やした人はそれなりに変更してもらいたい。あと機種別に変更が必要なのはリスト6の5～8行のラベルに定義している値だが，これについてはすでに述べた。

それから先月の501～504行をリスト7と差し換える。これは敵が爆発するようになったことによるジャンプテーブルの追加だ。キャラクタタイプの03_H～$2F_H$にあたる部分は，将来新しいキャラクタを追加するときのために空けてある。拡張の際には，この空いている部分に移動処理ルーチンの先頭アドレスを登録すればよいようになっているわけだ。

また，得点表示ルーチンの呼び出しも忘れちゃいけない。今月のリスト2aの22行と23行の間，それと先月のリスト5の264行と265行の間に，

CALL　SCOSUB

を挿入する。

さらにメインルーチンをリスト8のように変更する。変更点はゲームオーバー処理の追加，得点・残機の初期化ルーチン呼び出しの追加，トリガー1を押したらゲームが始まるようにしたこと，さらにゲームの速度を調節するために適当な時間稼ぎを入れたことだ。機種によってはリストどおりでは遅すぎたり速すぎたりするだろうから，39行の値を増減することで適当に調整してもらいたい。

あとは爆発のキャラクタパターンデータを付け加える。一応それらしいものをリスト9に用意してはみたが，自分で作ったものがあればそれを使うに越したことはない。キャラクタパターンはリストのどこに入れても構わないが，PCGパターンは順番に並べてあるわけだから，ちゃんと決められた場所に入れること。先月のリスト3の280行目を削除し，代わりに23行以降を挿入する。また，PCGを使う機種ではさっきも言ったように，$F0_H$～$F9_H$に数字のパターンを作って入れる。DEFSで隙間のバイト数を調節し直すのを忘れちゃだめだよ。10個のPCGを追加したわけだから，305行のDEFSで確保するバイト数は24（＝PCG1個あたりのバイト数）×10減るね。

ではアセンブルし，実行してみよう。半年の成果がこれだ。しばらく遊んだら，おもむろに調整・拡張を始めよう。

リスト9

```
 1 ;
 2 ;        爆発パターン（MZ-700）
 3 ;
 4 BMPAT1: DEFB    0F9H,0F6H,0F6H,0F9H        ;爆発1
 5         DEFB    061H,061H,061H,061H
 6 BMPAT2: DEFB    0F6H,0F9H,0F9H,0F6H        ;爆発2
 7         DEFB    061H,061H,061H,061H
 8 BMPAT3: DEFB    0F1H,0F2H,0F4H,0F8H        ;爆発3
 9         DEFB    061H,061H,061H,061H
10 ;
11 ;        爆発パターン（MZ-2000/2200）
12 ;
13 BMPAT1: DEFB    01EH,01EH,01EH,01EH        ;爆発1
14 BMPAT2: DEFB    01FH,01EH,01FH,01EH        ;爆発2
15 BMPAT3: DEFB    01FH,01FH,01FH,01FH        ;爆発3
16 ;
17 ;        爆発パターン（MZ-1500/2500/X1/X1turbo）
18 ;
19 BMPAT1: DEFB    0A6H,0A7H,0A8H,0A9H        ;爆発1
20 BMPAT2: DEFB    0AAH,0A7H,0ABH,0A9H        ;爆発2
21 BMPAT3: DEFB    0ACH,0ADH,0AEH,0AFH        ;爆発3
22 ;pcg
23         DEFB    0FCH,0F0H,0C0H,080H,080H,000H,000H,000H ;A6H
24         DEFB    003H,00FH,03FH,07FH,07FH,0FFH,0FFH,0FFH
25         DEFB    003H,00FH,03FH,07FH,07FH,0FFH,0FFH,0FFH
26         DEFB    01FH,007H,003H,001H,001H,000H,000H,000H ;A7H
27         DEFB    0E0H,0F8H,0FCH,0FEH,0FEH,0FFH,0FFH,0FFH
28         DEFB    0E0H,0F8H,0FCH,0FEH,0FEH,0FFH,0FFH,0FFH
29         DEFB    000H,000H,000H,080H,080H,0C0H,0E0H,0F8H ;A8H
30         DEFB    0FFH,0FFH,0FFH,07FH,07FH,03FH,01FH,007H
31         DEFB    0FFH,0FFH,0FFH,07FH,07FH,03FH,01FH,007H
32         DEFB    000H,000H,000H,001H,001H,003H,007H,01FH ;A9H
33         DEFB    0FFH,0FFH,0FFH,0FEH,0FEH,0FCH,0F8H,0E0H
34         DEFB    0FFH,0FFH,0FFH,0FEH,0FEH,0FCH,0F8H,0E0H
35         DEFB    0F8H,0E0H,0C0H,080H,080H,01EH,03FH,03FH ;AAH
36         DEFB    007H,01FH,03FH,07FH,07FH,0E1H,0C0H,0C0H
37         DEFB    007H,01FH,03FH,07FH,07FH,0E1H,0C0H,0C0H
38         DEFB    03FH,03FH,01EH,080H,080H,0C0H,0E0H,0F8H ;ABH
39         DEFB    0C0H,0C0H,0E1H,07FH,07FH,03FH,01FH,007H
40         DEFB    0C0H,0C0H,0E1H,07FH,07FH,03FH,01FH,007H
41         DEFB    0F8H,0E0H,0C7H,09FH,0BFH,07FH,0FFH,0FFH ;ACH
42         DEFB    007H,01FH,038H,060H,040H,080H,000H,000H
43         DEFB    007H,01FH,038H,060H,040H,080H,000H,000H
44         DEFB    01FH,007H,083H,0E1H,0F1H,0F8H,0FCH,0FCH ;ADH
45         DEFB    0E0H,0F8H,07CH,01EH,00EH,007H,003H,003H
46         DEFB    0E0H,0F8H,07CH,01EH,00EH,007H,003H,003H
47         DEFB    0FFH,0FFH,07FH,0BFH,09FH,0C7H,0E0H,0F8H ;AEH
48         DEFB    000H,000H,080H,040H,060H,038H,01FH,007H
49         DEFB    000H,000H,080H,040H,060H,038H,01FH,007H
50         DEFB    0FCH,0FCH,0F8H,0F1H,0E1H,083H,007H,01FH ;AFH
51         DEFB    003H,003H,007H,00EH,01EH,07CH,0F8H,0E0H
52         DEFB    003H,003H,007H,00EH,01EH,07CH,0F8H,0E0H
```

リスト10

```
;
;        仮想VRAM->実VRAM転送
;        （VRAMはD000H以降）
;
DISP700:
DISP:   LD      HL,4*48+4+AVRAMMODOKI
                                    ;HL=仮想AVRAM(4,4)のアドレス
        LD      DE,0D800H           ;D=実AVRAM(0,0)アドレス
        EXX
        LD      HL,4*48+4+VRAMMODOKI
                                    ;HL=仮想VRAM(4,4)のアドレス
        LD      DE,0D000H           ;D=実VRAM(0,0)アドレス
        LD      B,24
DISP0:  PUSH    BC
        LD      BC,40               ;実画面横サイズ
        LDIR                        ;横40文字分ブロック転送
        LD      C,8                 ;HL=HL+8
        ADD     HL,BC               ;
        EXX
        LD      BC,40               ;実画面横サイズ
        LDIR                        ;横40文字分ブロック転送
        LD      C,8                 ;HL=HL+8
        ADD     HL,BC               ;
        EXX
        POP     BC
        DJNZ    DISP0
        RET
```

MZ-700の色ずれ防止策

MZ-700版は色がずれて画面がチラチラしてしまっていた。これは，まずテキストVRAM全画面に書き込みを行い，続いてアトリビュートVRAMへ書き込むという処理の時間差のためだ。そこで，テキストVRAMとアトリビュートVRAMに1バイトずつ交互に書き込むように修正してみた。リスト10だ。仮想VRAMから実VRAMへの表示ルーチンDISPとそっくり入れ替えて使ってもらいたい。

ジョイスティックへの対応

このゲームで採用したキーボードからの入力データフォーマットは，元よりジョイスティックを意識していた。そういうわけだからジョイスティックからのデータ入力方法さえわかればジョイスティック対応にすることはわけない。

リスト11のa，bにX1とMZ-2500用のジョイスティック入力ルーチンを示す。詳しくはリストの注釈と文献4，5を参照してもらいたいのだが，簡単に解説しておくとしよう。まずはX1のほうから。

X1のジョイスティックはPSGを介して接続されている。サウンド用LSIとジョイスティックというのも妙な取り合わせだが，PSGという石が持っている汎用の8ビットデータ入出力機能をうまいぐあいに利用しているというわけだ。

PSGは内部にいくつかのレジスタを持っており，順に番号が付いている。このうち内部レジスタの14，15それぞれにジョイスティック1，2がつながっている。PSGの内部レジスタはI/Oポートの$1Bxx_H$に内部レジスタ番号をOUTしてから，$1Cxx_H$を通してアクセスする。ここで「xx」になっている部分はいくつでも構わないという意味だ。つまりBCレジスタで間接アドレッシングするときのCレジスタの値には気を使う必要がない。

こういう単純な仕組みなので，ジョイスティック1からの入力は$1Bxx_H$に14をOUTして，$1Cxx_H$からデータをINするだけですむ。が，PCGのレジスタ14，15は入力にも出力にも切り換えて使えるような設計になっていて，この切り換えをレジスタ7で行うことになっているから，あらかじめレジスタ14を入力用に設定しておかなければ意味のあるデータが帰ってこない。その辺の設定を行っているのがリスト6の6～12行だ。レジスタ7の第6ビットを0にしておけばレジスタ14が入力用になる。

続いてMZ-2500。MZ-2500のジョイスティックはI/Oポートの$F0_H$につながっている。ここにジョイスティック1と2のどちらから入力するか（など）のコマンドを送ってから，データを読み込むとジョイスティックの状態が返される。MZ-2500のジョイスティックからの入力データはX1のもの（＝ゲームで採用したキーデータ形式）とはちょっと違うので，リストでは適当にビット操作をしてデータ変換をしている。より具体的なことはオーナーズマニュアルを参照してもらいたい。

turboZ版仕様の追加

と言っても，大きな変更を加えようというのではない。turboZではテキストに64色中8色のパレットが使えるのを思い出したので，テキストパレットの設定ルーチンを付けよう，というただそれだけの話だ。リスト12をマージしたうえで，先月のリスト4の8行目の下にでも，

CALL　DEFTPL

の1行を追加してもらいたい。設定する色はTPLTBL以下にパレットコード1～7の順に並んでいる。いまは多少原色をくすませた程度の色になっているから，適当に変更するのがよいだろう。

turboZのテキストパレットの設定は，まずマルチモードにしてから，I/Oポートの$1FB9_H$～$1FBF_H$（順にカラーコード1～7に対応）に64色のカラーコードをOUTすることで行う。設定が終わったらコンパチモードに戻しておいたほうがよいだろう。

マルチモードとコンパチモードはI/Oポート$1FB0_H$の第7ビットで切り換える（1のときマルチモード，0のときコンパチモード）。このポートは画像取り込みなどにも関係しているので，単純にモードを切り換えたいだけなら，第0～6ビットは0にしておくようにする。

イベントの追加

いまのところこのゲームはイベントデータ列がなくなった時点でプツンと終わるようになっている（注4）。イベントデータを増やせばいくらでもゲームを引き伸ばすことは可能なのだが，ここではイベントデータを繰り返し使うことを考える。それには，「イベントデータ列へのポインタを初期化する（先頭を指すように戻す）イベント」を用意すればよい。先月，ろくに解説もせずにリストに忍び込ませてあったイベント番号80_H，「マップデータ列へのポインタを初期化するイベント」のようなものだ。

リスト13ではイベント番号81_Hに「イベントデータポインタを初期化するイベント」を割り当ててあり，このイベントは同時にマップデータポインタの初期化も行うよう

リスト11

a

```
;
;        ジョイスティックから入力
;        （キー入力ルーチンと差し換えて使う）
;
JOYGETX1:
INKEY:   LD      B,1CH            ;BC=1CxxH
         LD      A,7              ;PCGレジスタ7を
         OUT     (C),A            ;   セレクト
         DEC     B                ;BC=1BxxH
         IN      A,(C)            ;PCGレジスタ7の
         AND     0BFH             ;   第6ビットをリセットして
         OUT     (C),A            ;   出力し直す
         INC     B                ;BC=1CxxH
         LD      A,0EH            ;PCGレジスタ14を
         OUT     (C),A            ;   セレクト
         DEC     B                ;BC=1BxxH
         IN      A,(C)            ;   ジョイスティックデータ入力
         RET
```

b

```
;
;        ジョイスティックから入力
;        （キー入力ルーチンと差し換えて使う）
JOYGET2500:
INKEY:   LD      A,2FH            ;00101111B
                                  ;ジョイスティック1有効
                                  ;ジョイスティック1を選択
         OUT     (0EFH),A         ;
         IN      A,(0EFH)         ;ジョイスティックから入力
         LD      B,A              ;ゲームで使う
         ADD     A,A              ;   キーデータフォーマットに
         AND     0F0H             ;   変換する
         LD      C,A              ;
         LD      A,B              ;
         AND     0FH              ;
         OR      C                ;
         RET
```

リスト12

```
;..................................................................
;
;          テキストパレット設定（turboZ）
;
DEFTPL: LD      A,80H           ;マルチモードへ
        LD      BC,1FB0H        ;
        OUT     (C),A           ;
        LD      HL,TPLTBL       ;パレットデータ
        LD      BC,1FB9H        ;I/Oアドレス
        LD      E,7             ;ループカウンタ
INITP0: LD      A,(HL)          ;
        OUT     (C),A           ;
        INC     HL              ;
        INC     BC              ;
        DEC     E               ;7色分
        JR      NZ,INITP0       ;  繰り返す
        XOR     A               ;コンパチモードへ
        LD      BC,1FB0H        ;
        OUT     (C),A           ;
        RET
;
TPLTBL: DEFB    2BH,2EH,2FH,3AH ;1-4
        DEFB    3BH,3EH,2AH     ;5-7
```

リスト13

```
;
;          イベントジャンプテーブル
;
IJMPTBL:
         DEFW     IVNT80,IVNT81,DMY,DMY    ;80-83
         DEFW     DMY,DMY,DMY,DMY          ;84-87
         DEFW     DMY,DMY,DMY,DMY          ;88-8B
         DEFW     DMY,DMY,DMY,DMY          ;8C-8F
;
;          イベントno.81H
;          イベントデータ/マップデータを繰り返す
;
IVNT81:  LD       HL,IVNTDAT+1     ;イベントデータポインタの
         LD       (IVNTP),HL       ;          初期化
;
;          イベントno.80H
;          マップデータを繰り返す
;
IVNT80:  LD       HL,MAPDAT        ;マップデータポインタの初期化
         LD       (MAPPNT),HL      ;
DMY:     OR       A
         RET
```

リスト14

```
;
;          敵タイプA（直進）移動処理
;
TEKIA:   LD       A,(IX+X)                ;X座標が
         CP       40                      ;  40ちょうどであれば
         CALL     Z,EFIRE                 ;  弾を撃つ
         RRC      (IX+WORK1)              ;4回につき3回だけ進む
         JP       C,PUT                   ;移動しなくても表示はする
         XOR      A                       ;A=向きコード0（左へ）
         JP       MOVE&PUT                ;移動&表示
;
ADAT1:   DEFB     0,44,10                 ;イベント1
         DEFW     10*48+44+VRAMMODOKI
         DEFW     CPATA
         DEFB     88H,0,0                 ;←ここを直す
;
ADAT2:   DEFB     0,44,20                 ;イベント2
         DEFW     20*48+44+VRAMMODOKI
         DEFW     CPATA
         DEFB     88H,0,0                 ;←ここを直す
```

になっている。また，先月のマップデータポインタ初期化イベントは余計なことをしていたようなので，ついでにいらない部分を削ってある。

このマップデータとイベントデータを繰り返し使うためのイベントは，マップデータとイベントデータをうまく作らなければ意味をなさない。先月のリスト中の注釈に示したように，マップデータは1画面分が60バイト(横10×縦6)，イベントデータは1画面分が40バイト（1キャラクタスクロールするたびにイベントが発生する）になっている。仮に背景を2画面分用意することを考えると,「マップデータポインタを初期化するイベント」も正確に2画面分ごとに入れなければならない。また，このマップの継ぎ目を見せないようにするために，マップデータの最後には，最初の1画面とまったく同じデータを付けておかなければならないことがわかる（正確には1画面+1列が必要)。

さらにイベントデータポインタを初期化するイベントも，背景とずれないようにするために，背景の画面数の倍数の位置に入れることになる。

これらについては，あとで示すサンプルデータも参考にしてほしい。

注4）あんまりプツンと終わるので暴走したと勘違いした人もいるとかいないとか。これはちょっと考えてみればわかることだが，このゲームではS-OS“SWORD”などのシステムサブルーチンを利用せずに画面のモードを切り換えて使っているので，プログラムが終了したあとの画面はゲームで設定したままの状態になっている。そのため，文字が正しく表示されないといったことも起きるわけだ。画面を元に戻すには，システムの画面初期化コマンド（“SOWRD”なら#W)を実行するか，リセットするかしてもらいたい。

直進タイプの改造

これまで直進タイプの敵キャラは，ただボーっと真っ直ぐに飛んでくるだけだったが，今回は弾も撃たせてあげようと思う。リスト14だ。X座標が40のときに1発弾を撃つようにし，ついでに移動速度を3/4に落としてみた。つまりメインループ4回のうち3回移動するようにしたわけだ。これには例のローテート式カウンタを使っている。背景のスクロール速度を落とすのに使ったあの方法だ。

スクロールのときはカウンタを88_Hというように，4ビットごとに1になるようなデータで初期化しておき，RRCで回転させながらキャリフラグによってスクロールするかしないかを決めたわけだ。これによりスクロール速度は1/4になった。今度は3/4だからEE_Hのように4ビットのうち3ビットが1になっている値をカウンタの初期値にすればよいだろう。また，88_Hのように4ビットのうち3ビットが0になっているデータを使って条件判断を逆にしても同じことだ。リストでは後者の方法を使っている。カウンタにはWORK1を使用した。リストの注釈に示したように，キャラクタが登場する際の初期データも変更しなければ意味がない（カウンタの初期値を設定しなくちゃ）ことに注意してもらいたい。

動かない敵を加える

次に敵の種類をちょっと増やしてみよう。拡張キャラクタ第1弾は天井にへばりついて動かない，つまり，背景のスクロールと連動して動く敵キャラだ。

これは背景スクロールのカウンタを覗き見て，直前にスクロールが行われたかどうかを調べ，それに合わせて左に動くようにすることで簡単に実現できる。リスト15はこの固定タイプの敵キャラの移動ルーチンだ。リストではスクロールのカウンタを調べるのではなく，同じ速さで変化する別のカウンタを自分のデータバッファ内に用意している。

この新しいキャラクタをプログラムに組み込むには，まず，移動ルーチンの先頭アドレスをジャンプテーブルに登録する。3番目のキャラクタということで，キャラクタタイプコード2（0から数えている）を割り当てることにすると，リスト7の1行目の3番目に並んだDMYをTEKICに書き換えればよい。

移動ルーチンを組み込んでも，登場するイベントがなければ出てこられないので，次にイベントコードを割り当てる。いまの

ところ敵の登場イベントは1〜5を使っていたから6以降が使える。リストでは初期位置のY座標が違う2種類のイベントを用意し，これにイベントコード6，7を割り当てた。そして登場時の初期データを作成し，いままでの初期データの最後に加える。リスト15の15行〜23行を先月のリスト5の461行以降に挿入しよう。

また，イベントの数が2つ増えたことにより，先月のリスト5の400行のループカウンタの値に2を足して7とする。それからキャラクタパターンを作り，PCGを使う場合はPCGデータも追加する。キャラクタパターンとPCGパターンは27行以下に用意してはみたのだが，あんまりカッコよくないので，自作のキャラクタに差し換えてくれるとありがたかったりする。

加えて，得点のテーブルにそのキャラクタの点数を登録し，最後にイベントデータにこの新しいイベントを混ぜれば組み込み終了だ。ただ，この固定タイプは背景と一致させる必要があるので，マップデータとのつじつまも合わせておかなければならない。

空から降ってくる敵もいる

拡張キャラクタ第2弾として，画面の上から落下してくるキャラクタを作ってみた。このキャラクタは自分の足元に障害物がなければ真下に落ち，障害物があればそれに沿って横に進むというものだ。また，横に進んでいる途中で壁に突き当たったら180度ターンすることにした。しかもターンするときに弾を1発撃つようになっているので，床にくぼみを作っておいて，そこにはまるようにイベントデータを作ると，往復しながらどんどん弾を撃ってくるという可愛らしさだ。

リスト16がこの落下タイプの移動処理ルーチンおよび登場時の初期データだ。現れる位置と左右の進行方向の異なる2種類のイベントを用意し，それぞれイベントコード8，9を充ててある。また，キャラクタタイプコードは3だ。

移動処理ルーチンは，リスト中の注釈を参考にすればなにをやっているかはすぐにわかるだろう。自機と障害物との当たり判定用に用意したサブルーチンISWALLで自分の足元や進行方向が壁かどうかチェックしている。

リスト15

```
 1 ;
 2 ;          敵タイプC（固定）移動処理
 3 ;
 4 TEKIC:   RRC     (IX+WORK2)        ;4回に1回しか
 5          JP      NC,PUT            ;  進まない
 6          XOR     A                 ;向きコード0（左）
 7          CALL    MOVE              ;移動
 8          RET     C                 ;場外に出た
 9 TEKIC0:  RRC     (IX+WORK1)        ;間隔をおいて
10          CALL    C,EFIRE           ;  弾も撃つ
11          JP      PUT               ;表示
12 ;
13 ;          追加イベントデータ
14 ;
15 ADAT6:   DEFB    2,44,8            ;イベント6
16          DEFW    8*48+44+VRAMMODOKI
17          DEFW    CPATC
18          DEFB    80H,22H,0
19 ;
20 ADAT7:   DEFB    2,44,12           ;イベント7
21          DEFW    12*48+44+VRAMMODOKI
22          DEFW    CPATC
23          DEFB    08H,22H,0
24 ;
25 ;          キャラクタパターン
26 ;
27 CPATC:   DEFB    04EH,04DH,042H,056H          ;MZ-700
28          DEFB    017H,017H,061H,061H
29 CPATC:   DEFB    01EH,01EH,098H,096H          ;MZ-2000
30 CPATC:   DEFB    0B8H,0B9H,0BAH,0BBH          ;MZ-1500/2500/X1
31 ;pcg
32          DEFB    0FFH,0FFH,0FFH,0FFH,0E0H,0FFH,0FFH,0FFH ;B8H
33          DEFB    00EH,00EH,00EH,00EH,000H,00EH,00EH,00EH
34          DEFB    03FH,03FH,03FH,01FH,000H,01FH,03FH,03FH
35          DEFB    0FFH,0FFH,0FFH,0FFH,007H,0FFH,0FFH,0FFH ;B9H
36          DEFB    000H,000H,000H,000H,000H,000H,000H,000H
37          DEFB    0C4H,0C4H,0C4H,0C8H,000H,0C8H,0C4H,0C4H
38          DEFB    0E0H,0E0H,0E0H,0E0H,0F2H,0F2H,0F9H,0FCH ;BAH
39          DEFB    01FH,01FH,01FH,01FH,00FH,00FH,007H,003H
40          DEFB    01CH,01CH,01EH,01FH,00FH,00FH,007H,003H
41          DEFB    007H,007H,007H,007H,00FH,00FH,01FH,03FH ;BBH
42          DEFB    0F8H,0F8H,0F8H,0F8H,0F0H,0F0H,0E0H,0C0H
43          DEFB    038H,038H,078H,0F8H,0F0H,0F0H,0E0H,0C0H
```

リスト16

```
;
;          敵タイプD（落下）移動処理
;
TEKID:   LD      L,(IX+ADRL)       ;表示アドレスを
         LD      H,(IX+ADRH)       ;  取り出す
         LD      DE,48*2           ;足元を指すように
         ADD     HL,DE             ;  補正する
         CALL    ISWALL            ;左下は壁か?
         JR      C,TEKID0          ;そうなら水平移動
         INC     HL                ;
         CALL    ISWALL            ;右下は壁か?
         JR      C,TEKID0          ;そうなら水平移動
         LD      A,(IX+Y)          ;落下しながら
         CP      8                 ;  Y座標が8の時に
         CALL    Z,EFIRE           ;  弾を1発撃つ
         LD      A,6               ;向きコード6（真下）
         JP      MOVE&PUT          ;移動・表示
TEKID0:  RRC     (IX+WORK2)        ;水平移動は
         JP      C,PUT             ;  1/2の速度にした
         LD      L,(IX+ADRL)       ;表示アドレスを
         LD      H,(IX+ADRH)       ;  取り出す
         DEC     HL                ;左正面
         LD      A,(IX+WORK1)      ;A=0なら左向きelse右向き
         LD      BC,0004H          ;B=左,C=右の向きコード
         OR      A                 ;A=0?
         JR      Z,TEKID1          ;そうなら左向き
         INC     HL                ;右正面
         INC     HL                ;
         INC     HL                ;
         LD      BC,0400H          ;B=右,C=左の向きコード
TEKID1:  CALL    ISWALL            ;進行方向に壁があるか?
         JR      C,TEKID2          ;そうならターン処理へ
         LD      DE,48             ;進行方向下
         ADD     HL,DE             ;
         CALL    ISWALL            ;進行方向下は壁か?
         JR      NC,TEKID3         ;そうでなければそのまま移動
         ;
TEKID2:  LD      A,(IX+WORK1)      ;向き反転
         CPL                       ;
         LD      (IX+WORK1),A      ;
         LD      B,C               ;向きコードも反転
         PUSH    BC                ;
         CALL    EFIRE             ;ついでに弾を1発
         POP     BC                ;
TEKID3:  LD      A,B               ;B=現在の進行方向の向きコード
         JP      MOVE&PUT          ;移動・表示
;
;          追加イベントデータ
;
ADAT8:   DEFB    3,36,2            ;イベント8
         DEFW    2*48+36+VRAMMODOKI
         DEFW    CPATD
         DEFB    0,88H,0
;
ADAT9:   DEFB    3,4,2             ;イベント9
         DEFW    2*48+4+VRAMMODOKI
         DEFW    CPATD
         DEFB    -1,88H,0
;
;          キャラクタパターン
;
CPATD:   DEFB    04EH,04DH,042H,056H          ;MZ-700
         DEFB    021H,021H,067H,067H
CPATD:   DEFB    097H,095H,01EH,01EH          ;MZ-2000
CPATD:   DEFB    0BCH,0BDH,0BEH,0BFH          ;MZ-1500/2500/X1
;pcg
         DEFB    0FCH,0F8H,0F0H,0F0H,0E0H,0E1H,0E3H,0E3H ;BCH
         DEFB    003H,007H,00FH,00FH,01FH,01FH,01FH,01FH
         DEFB    000H,000H,000H,000H,000H,000H,000H,000H
         DEFB    03FH,09FH,04FH,04FH,007H,087H,0C7H,0C7H ;BDH
         DEFB    0C0H,0E0H,0F0H,0F0H,0F8H,0F8H,0F8H,0F8H
         DEFB    000H,080H,040H,040H,000H,000H,000H,000H
         DEFB    0FFH,0FFH,0FFH,0E0H,0FFH,0FFH,0FFH,0FFH ;BEH
         DEFB    000H,000H,000H,000H,000H,000H,000H,000H
         DEFB    023H,023H,013H,000H,013H,023H,023H,023H
         DEFB    0FFH,0FFH,0FFH,007H,0FFH,0FFH,0FFH,0FFH ;BFH
         DEFB    070H,070H,070H,000H,070H,070H,070H,070H
         DEFB    0FCH,0FCH,0F8H,000H,0F8H,0FCH,0FCH,0FCH
```

仕上げは最終面のマップデータ

拡張はこのくらいにして，読者がデータを作るときの参考になるようなマップデータをサンプルとして載せておく。リスト17だ。

このゲームは現在のところ全1面だ。全部で1面ということは，これは最終面でもある(!)。リスト17はこのゲームの最終面ということになる。背景データは3画面分，イベントデータは6画面分だけしかない。詰め込んだので，敵が次々登場し，弾がバシバシ乱れ飛ぶという最終面にふさわしい内容になっている。

これでプレイしてみると，敵の撃った弾が壁をすり抜けて飛んでくることにア然とするだろう。そーゆー人は自弾と障害物の当たり判定ルーチンを参考に，敵の弾と障害物との当たり判定を行うサブルーチンを組み込んでもらいたい。また，敵が壁を通り抜けるという現象も見られるだろう。これは悪い見本であって，本当は「敵の軌跡と壁が重ならないように」マップデータとイベントデータを作らなければならないところだ。

ま，これはあくまで見本ということで，最終的にはマップもキャラクタも自作してひとりで楽しむなり，友だちにやらせるなりしてほしいと思う。もちろん，敵キャラの種類を増やしてみたり，アイテムを付け加えたり，タイトルを付けたり，効果音を加えたりといった改造もバンバンやってもらいたいものだ。

最後になったが，このゲームはいかなるシステムにも依存していない。つまり，アドレスさえ重ならなければBASICやCP/Mと共存することもできるわけだ。もちろん，ソースリストがあるわけだから，プログラムを置くアドレスはどこにでも変更できる。タイトルはBASICで表示し，このゲームを呼び出し，戻ったらゲームオーバーの画面を表示するといったことなら簡単に行えるはずだ。

とにかく，プログラム的にはいくつか手抜きはあったとしても，シューティングゲームの基本要素はみんな詰め込まれている(と思う)。こいつをベースに，あーでもないこーでもないといじくり回せば，しばらく遊べるんじゃないのかな。

リスト17

```
;
;          イベントデータテーブル  40バイトで1画面分
;
IVNTDAT:
          DEFB    1,1,0,0,2,2,0,0,3,3,3,3,0,0,4,4
          DEFB    4,4,0,0,5,5,5,5,0,0,0,0,1,1,0,0
          DEFB    2,2,0,0,3,3,3,3
          DEFB    0,0,4,4,4,4,0,0,5,5,5,5,0,0,0,0
          DEFB    1,1,0,0,2,2,0,0,3,3,3,3,0,0,4,4
          DEFB    4,4,0,0,5,5,5,5
          DEFB    0,0,4,4,4,4,0,0,5,5,5,5,0,0,0,0
          DEFB    1,1,0,0,2,2,0,0,6,0,0,0,6,5,5,5
          DEFB    6,0,0,0,6,0,0,2
          DEFB    80H,0,0,0,7,0,0,0,2,2,2,2,7,0,0,0
          DEFB    5,5,5,5,0,0,0,0,0,0,0,0,0,0,0,0
          DEFB    8,0,0,0,0,0,0,0
          DEFB    8,0,8,0,8,0,8,9,8,9,0,0,0,0,0,0
          DEFB    0,0,0,0,0,0,0,0,0,0,0,0,1,1,2,1
          DEFB    1,1,2,1,0,0,0,0
          DEFB    0,0,4,4,4,4,0,0,5,5,5,5,0,0,0,0
          DEFB    1,1,0,0,2,2,0,0,3,3,3,3,0,0,4,4
          DEFB    4,4,0,0,5,5,5,5
          DEFB    81H      ;イベントデータを繰り返すイベント
;
;          マップデータテーブル    60バイトで1画面分
;          (繰り返し使う時は最初と最後の
;                          ブロックを同じにしておく)
;
;
MAPDAT:   DEFB    0,0,0,1,0,4,2,2,2,2,2,4 ;
          DEFB    0,0,0,0,0,4,2,2,3,2,2,4
          DEFB    0,0,0,0,0,4,2,2,2,2,2,4
          DEFB    5,1,0,0,1,4,5,2,2,2,2,4
          DEFB    5,0,0,0,0,4,5,2,2,3,2,4
          DEFB    5,0,0,0,5,5,5,5,2,3,2,5 ;
          DEFB    5,0,0,0,5,5,5,5,2,2,2,5
          DEFB    0,0,0,0,0,5,1,2,2,2,2,5
          DEFB    0,0,0,0,0,5,2,3,3,2,2,5
          DEFB    0,0,0,0,0,5,2,2,2,2,2,5
          DEFB    0,0,5,0,0,4,2,2,2,2,2,4 ;
          DEFB    0,0,0,0,0,4,2,2,2,2,2,4
          DEFB    0,0,0,0,0,5,2,2,2,2,2,4
          DEFB    0,0,0,0,0,4,2,2,2,2,2,4
          DEFB    0,0,0,0,0,4,2,2,2,2,2,4
          DEFB    0,0,0,1,0,4,2,2,2,2,2,4 ;
          DEFB    0,0,0,0,0,4,2,2,3,2,2,4
          DEFB    0,0,0,0,0,4,2,2,2,2,2,4
          DEFB    5,1,0,0,1,4,5,2,2,2,2,4
          DEFB    5,0,0,0,0,4,5,2,2,3,2,4
          DEFB    5,0,0,0,5,5                     ;
```

最後のご挨拶

ということで，「Z80マシン語ゲーム工房」は無事完結した。いま読み直してみると，「ほどよい説明と説明不足」，「ハイペースと消化困難なページ数」の間でキワどいバランスを保ちながらも，どちらかっていうとよい方向に転んだんじゃないかと思う。僕にしてはまずまずのデキだ。適度に既存の「ましんごにゅーもん」のパロディになっている部分もあったりなんかするし。

この連載は「ゲーム作りによるマシン語講座風」にしたつもりでいる。「マシン語によるゲームの作り方講座」の色合いが強いように思われたかもしれないが，やっぱり「マシン語入門」だったのだ，と思いたい。僕が書きたかったのはゲームのアルゴリズムなんかではなく，マシン語でプログラムを作ることの楽しさみたいなものだった。入門者クラスの人にはいくつかの命令をパズルのように組み合わせること自体が楽しいのだということを，もうちょっと上のレベルの人にはデータ構造とかプログラムの構造の設計をすることが楽しいのだということを，また，ハードを直接いじって，マシンを思いどおりに動かすことが楽しいのだということを感じてもらいたかったのだ。

これらの楽しさに触れた人は，きっといまごろ自分でクイズを作って考えていたり，オリジナルの敵キャラのアルゴリズムを考えたり，効果音を付けるためにPSGの資料を調べたりしていることだろう。

すべての読者が最初から最後まですんなり読み通せるとは思っていない（僕は説明がヘタだからね)。それでも，自分で考え，試してみることを怠らなければ，道はそれほど険しくはないんじゃないだろか（ああ，歯が浮く)，と考えている。今月は新しいアセンブラが発表されるし，いままでアセンブラがなくて傍観していた人も，また最初から挑戦してもらいたいと思う。読んだだけでわかったつもりになっちゃうのが一番怖いからね。

それでは，この得体のしれない記事に最後までお付き合いいただいた皆さんにめいっぱい感謝しつつ，また，別の機会までご機嫌よう，なのでした。

〈参考文献〉

1) MZ-700オーナーズマニュアル，シャープ
2) MZ-1500オーナーズマニュアル，シャープ
3) MZ-2000オーナーズマニュアル，シャープ
4) MZ-2500オーナーズマニュアル，シャープ
5) 祝一平，試験に出るX1，日本ソフトバンク
6) MZ-2500テクニカルマニュアル，工学社

とおりゃんせなのである

Iwai Ippei
満開製作所 祝 一平

先月までCを使ってピコピコとゲームで遊んでいたこの講座も，新しい年の幕開けとともに気分も一新，突如としてフィルタを作ってしまうという実用講座へと変身を遂げてしまったよーです。オマケに最後には皆さんへの宿題まで用意してあるという，気合の入った超豪華版(？)を今月はお届けします。

前回までのピコマゲドンは，なかなかにすっぽんであったが，今月は反省しつつ，それなりに実用を目指してフィルタを作ってみたりするのである。

さて，「フィルタ」とはなんぞや，などといぶかる向きもあるかもしれないが，簡単にいってしまえば，フィルタとは「標準入力からデータを持ってきて，標準出力に出す」というプログラム(コマンド）のことである。でもって，その途中でデータに対してあれこれするのがお仕事なわけだな（この論でいくと，1988年8月号のultra.cは「なんにもしないフィルタ」だったわけである)。

まずはリスト1のyasa.cに，うんと簡単な機能のフィルタを示す。このプログラムの機能は,「指定された文字を別の指定された文字で置き換える」ものである。文字列ではなく，あくまで1バイトの文字を置き換えるだけであることに注意していただきたい。試しに，

　　yasa a ＊ ＜ yasa.c

などとすると，「a」が「＊」に置き換えられて，

　　m＊in (＊rgc, ＊rgv)
　　int ＊rgc;
　　　⋮

というように表示されるであろう。「＜」は前にもやったI/Oリダイレクションであるから，思い出していただきたい。

では，説明しよう。標準入力から入力するのはgetchar ( ) なわけである。で，それを変数cに代入して，「＊argv[1]」と同じかどうかをif文でチェックする。違っているのなら，putchar (c)で，cをそのまま標準出力に出す。もしも同じだったなら，＊argv[2] をputchar(～) によって標準出力に出す。これで置き換えになるわけだ。このよーなことを，EOF（エンド・オブ・ファイル）になるまで続ける。以上，きわめて単純なプログラムである。

てなわけで，ひとまず問題の核心は，＊argv[1] と＊argv[2]に移っていくのであった。

リスト1　yasa. c

```
#define EOF     -1

main(argc,argv)
int argc;
char *argv[];
{
        int c;

        while ((c = getchar()) != EOF)
                if (c == *argv[1])
                        putchar(*argv[2]);
                else
                        putchar(c);
}

```

プログラムに渡すのこと

すでにお馴染みになっているのではないかと思うが，Cで書かれた多くのプログラムには，

　　main (argc, argv)
　　int argc;
　　char ＊argv[ ];

なんてのがあることになっている（最近のANSI 準拠のCではちょっと違う書式になっているようだけど)。これはプログラムの起動時に引数を与えるときの書式なのだが，引数群は文字列として順に配列に格納される。このときの引数の個数がargc，配列のポインタがargvである。で，このよーにしておいたプログラムをmr.cとしよう。そして，それをコンパイルしてできたプログラムがmr.xなわけだ。ここで，

　　A＞mr uumu tiger me![CR]

なんてのを打ち込むと，mr.xが起動される。このとき，mr.cのなかの変数argcには，自動的に「4」が入っているのである。また，

　　argv[1] には“uumu”のアドレス
　　argv[2] には“tiger”のアドレス
　　argv[3] には“me!”のアドレス

が入っている。ちなみに

　　argv[0] には“A:¥mr.x”のアドレス

が入っていたりするので，なかなか便利であろう（「A:¥」というのは，たまたまmr.xがドライブAのルートディレクトリにあったからであることに注意)。

ここでスルドイ人は，「argv[3]までなのに，どーしてargcは4なんでい」と思うであろう。しかしこれは，Cができたときから，そーなっているんだ，ということで，とりあえず納得していただきたい。

で，リスト2がmr.cなわけである。これで，引数(さっきの“uumu”とかである）の渡され方をねっとりと確認することができる。プログラムは渡された引数を順に，長さ，16進数でのダンプ，実物，というぐあいに表示するものである。こまごまとした実情については，実際にこのようなプログラムを使って，シコシコと確かめるのがいちばんよいのである。次ページに例を載せておく。

話は変わるが，argcとargvは，実は別にこのような変数名である必要はないのである。だからして，mainの宣言のなかで，

main(c, v)

とでもしておけば，それはそれでよいのである。そして，キーを叩く数が減って経済的なのであるが，しかし，歴史的事実としてargc，argvが使われてきたので，ハナモゲラ＋面倒でも，やはりこのままで置いておくべきであろう。

引数の規則

mr. xを使っていくと，いくつかの事実がわかる。

1) ダブルクォート「“」で囲まれた文字列は，ダブルクォートを取り除いて引き渡される。これによって，スペースを含む文字列をプログラムに渡すことができる。

2) シングルクォート「‘」で囲まれた文字列も，シングルクォートを取り除いて引き渡される。これによって「“」を含む文字列を渡すことも可能となる。逆もまた真で，ダブルクォートで囲むことによって，「‘」を文字列中に含ませることができる。

3) ただし，その結果「“‘」の両方が含まれた文字列は不可能である。

4) コントロールコードも渡せる（[ESC]＋コントロールコードで，^M＝CRなども可能）。ただし^Iは“^I”とする必要がある。また，^@は00_H，つまり文字列終端記号であるから，渡せない。

ありがちなバグ

1文字だけではなく，ちゃんと文字列を置換できるようにしたのがリスト3のmuzuka. cである。このプログラムは，一見すると動きそうに見えるし（わりと複雑だから一見じゃ，ちと無理かな），実際にいろいろやってみてもキチンと文字列を置換しているよーに見える。ところがどっこい，これにはバグがあるのである。どーゆーことかというと，たとえば，

123→ABC

として置換しようとしたとき，「11234」の入力を「1ABC4」にするはずなのに，実際は「11234」と，素通ししてしまうのである。これは，

1→“1”だ。一致してるのかな？

1→“2”じゃないから不一致だ。また始めからチェック。

2→“1”じゃないから不一致

3→“1”じゃないから不一致

4→“1”じゃないから不一致

（結局一度も一致しなかった）

という次第なのである。つまりは，「途中で不一致になった場合」は，あと戻りしなければならないのだ。これを忘れると，たちまちズルムケるわけである。そしてさらに，「123123123」なんてのを入力してやると，「ABC123ABC」なんて出力になってしまう。これは，入力から1文字読みすぎてしまうからなのであった（“1231”と“ABC\0”が対応してしまっているので，次のチェックを“2”から始めてしまうのだな）。そしてそしてさらにさらに，漢字を扱えないという症状も発見してしまった。これは，シフトJISコードに80_Hよりも大きい値が含まれているからである（値がマイナスということになってしまうので，“＝＝”で比較したときにヒネてしまう）。昔の宣教師もいってたが，やはり日本語は悪魔の言葉であることよ。以上，初心者および，いーかげんな所長が陥りやすいワナであった。

最後にいっておくが，大事なのは，どんな場合でもファイルエンド（EOF）に対応できるようにしておくことである。さもないと，FF_Hを出力してしまうことになる。リスト3の22行がそこらへんに対応しているところである。

おっとどっこい

というわけで，最終的にできたのがリスト4のyarinao. cである。ここのミソはungetchar2という関数である。これは，標準入力からの入力をバッファリングする関数である。早い話が，一度読み込んだデータを（疑似的に）戻すことができるようにした関数である。しかし，本当に戻しているわけではないので，getchar2( )という関数を作って，それを使って読み出さなければならない。

説明すると，この“char 2 システム”を利用するには，まずはinitchar 2 ( )を呼び出して，バッファやカウンタを初期化しておかなければいけない。このとき，バッファへのポインタと，そのバッファの大きさ（バイト数）も知らせてやるわけだな。リスト4ではとりあえず96バイトの配列を用意して，それを使っている。16，19行の“& 0xff”は，char 型の値が80_Hよりも大きい場合は，intに拡張されるとマイナスの値になってしまうので，それを防ぐためである。そして24，27行でEOFに対応していることにも注意。また33行では，変数iをデクリメントしながらungetchar 2 ( )

リスト2　mr. c

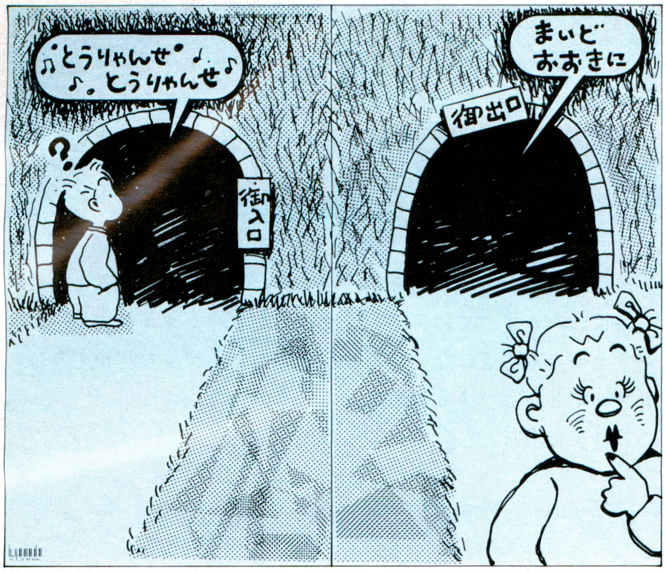
```
main(argc,argv)
int argc;
char *argv[];
{
        int i,j,l;

        for(i=0;i<argc;i++) {
                l = strlen(argv[i]);
                printf("%d=%d:",i,l);
                for(j=0;j<l;j++)
                        printf(" %02X",argv[i][j] & 0xff);
                printf(":%s¥n",argv[i]);
        }
}

```

図1　mr. x実行例

```
A>
A>mr uumu tiger me!
0=7: 41 3A 5C 6D 72 2E 78:A:¥mr.x
1=4: 75 75 6D 75:uumu
2=5: 74 69 67 65 72:tiger
3=3: 6D 65 21:me!
A>
A>
A>mr doughnut WARUINOKA
0=7: 41 3A 5C 6D 72 2E 78:A:¥mr.x
1=8: 64 6F 75 67 68 6E 75 74:doughnut
2=9: 57 41 52 55 49 4E 4F 4B 41:WARUINOKA
A>
A>
A>mr ^A^A ^[^[
0=7: 41 3A 5C 6D 72 2E 78:A:¥mr.x
1=2: 01 01:
2=2: 1B 1B:
A>
A>
```



を呼び出していることを理解していただきたい。

プログラムはここでいちおう完成したので，どーゆーふうに使うかを少々例示しておこう。たとえば，データ中からすべての改行を取り除くのであれば，

yarinao ^J “” < file1 > file2

とすればよい。なお，しつこいようだが，^Jを打ち込むには，[ESC]+[CTRL-J]である。そして，さらにそのファイルから TAB を取り除きたいのならば，

yarinao “^I” “” < file2 >file3

である。また，文章中にあるすべての「。」のあとで改行したいのであれば，

yarinao ‘。’ ‘。^J’ < file1 > file2

となる（縁起物だからシングルクォートでくくってみた)。ただしこうなると，すでに「。」のあとに改行があった場合は，改行が2つつながってしまうことになる。そこで，

yarinao ‘。^J^J’ ‘。^J’ < file2 > file3

とすると若干解決するわけである。めでたし，めでたし。

で，このようにすると大きな文書ファイルを加工する際になかなかに便利である。たとえばED. X などでは，文字列の置換が異常に遅いので，その方面にも活用できるかもしれない。

ところでシフト JIS コードというものは2バイトコードなので，そこらへんの処理をしようと思ったらかなり大変なことになるので，その方面は手抜きしてある。必要な方は各自で対応してもらいたい。

さて，宿題である

ここで発展問題として，宿題を出すことにする。

「AS. Xで，マルチステートメントを使えるようにするためのフィルタを作成せよ」

である。ステートメントの区切りは「：」ということにしよう。もちろん，「：」があったなら，なにがなんでも2行にしてしまうのではだめである。なぜかというと，

1）文字列中とか，コメントに「：」を使えなくなってしまうからである。

さらには，

2）ラベルにコロンを付けられなくなってしまう（label：～というのがあるでしょ)

という問題も出てくる。

結局，プログラムの満たすべき条件としては，

1）文字データ中の「：」は無視する
2）文字データの初めを示すのは“もしくは‘である
3）“で始まった文字データは“もしくは改行で終了する
4）‘で始まった文字データは‘もしくは改行で終了する
5）英数字と＿だけで行頭と接続しているコロンは，ラベルの右

リスト3　muzuka. c

```
#define EOF      -1

main(argc,argv)
int argc;
char *argv[];
{
        int c;
        int i,j;

        if (argc < 3) exit();

        while((c = getchar()) != EOF) {
                if (c == argv[1][0]) {  /* 先頭をチェック */
                        i = 0;          /* カウンタを初期化する */
                        while((c = getchar()) == argv[1][++i]);
                                        /* 一致している間中読み続ける */
                        if (argv[1][i] == '¥0')         /* NULLで不一致か？ */
                                puts(argv[2]);          /* それなら置き換え */
                        else
                                for(j=0;j<i;j++)        /* さもなくば元どうり */
                                        putchar(argv[1][j]);
                        if (c == EOF) break;    /* ファイルエンドならお終い */
                        putchar(c);             /* 最後に読んだ1字も忘れずに */
                } else
                        putchar(c);
        }
}

puts(s)
char *s;
{
        while(*s)
                putchar(*s++);
}
```

端に付いたコロンであるから，無視する

注）label：rts：nop

なんてのもあるから。

6）　行頭の「＊」，もしくはスペースかタブの直後の「＊」があったなら，その右側はコメントである（ただし文字データ中を除く）

7）　そのうえでマルチステートメントの「：」を，[改行]＋TAB（＝$0A_H$＋09_H）に変換する

若干ここで補足しておく。AS.Xは，「＊」がなくてもオペランド以降をコメントとして認めてしまうらしいので(たとえば「moveq ＃1,d0 comment」なんてのが通ってしまう)，本当はそこがコメントかどうかの判定には，きちんと文法解析しなければならないのであるが，問題がある場合はエラーが出るだろうから，適当に手直しすればよいだろうという，いーかげんな方針で迫っていただきたい。

最後にもうひとつだけ問題がある。それはprnファイルである。途中でマルチステートメントを2行（以上）に分割するのであるから，prnファイルに関しても清く正しいとはいえなくなってしまうであろう。

常々思っているのであるが，ZEDAなど，一部にあるように，この世にあるすべてのアセンブラはしかるべくマルチステートメントを使えるようになっているべきである。が，なぜかそうはハドソンが卸さないようである。あっち向いて，ぷん。

ではまた来月。

リスト4　yarinao. c

```
#define         EOF     -1

main(argc,argv)
int argc;
char *argv[];
{
        int c;
        int i,j;
        char buff[96];

        if (argc < 3) exit();

        initchar2(buff,96);

        while((c = getchar2()) != EOF) {
                if (c  == (argv[1][0] & 0xff)) {        /* 先頭をチェック */
                        i = 0;          /* カウンタを初期化する */

                        while((c = getchar2()) == (argv[1][++i] & 0xff));
                                        /* 一致している間中読み続ける */

                        if (argv[1][i] == '¥0') {       /* ＮＵＬＬで不一致か？ */
                                puts(argv[2]);          /* それなら置き換え */
                                if (c == EOF) break;    /* ファイルエンドなら終わり */
                                ungetchar2(c);          /* 戻すです */
                        } else {
                                if (c == EOF) { /* ファイルエンドで不一致？ */
                                        for(j=0;j<i;j++)        /* 吐き出して */
                                                putchar(argv[1][j]);
                                        break;          /* 終わり */
                                }
                                ungetchar2(c);          /* 最後に読んだ１字 */
                                for(i--;i>0;i--)        /* 元どおり */
                                        ungetchar2(argv[1][i]);
                                putchar(argv[1][0]);    /* ただし先頭は除外 */
                        }
                } else
                putchar(c);
        }
}

puts(s)
char *s;
{
        while(*s)
                putchar(*s++);
}

/* char2 works */

static char *char2_buff;
static int char2_max;
static int char2_count;

initchar2(p,max)                        /* バッファのイニシャライズなわけだ */
{
        char2_buff = p;                 /* set buffer address */
        char2_max = max;                /* set max size */
        char2_count = 0;                /* clear counter */
}

getchar2()
{
        if (char2_count)                /* バッファにあるか？ */
                return(char2_buff[--char2_count]);
        else
                return(getchar());
}

ungetchar2(c)
{
        if (char2_count<char2_max)      /* 空きはあるか？ */
                return(char2_buff[char2_count++] = c);
        else
                return(EOF);            /* 実際はチェックしていない */
}
```

ついに発売! OS-9/X68000

Komura Satoshi
古村 聡

昨年春のビジネスショウでお披露目になったOS-9がついに発売となりました。製品版の仕様・互換性はどうなったのか，編集室唯一のOS-9er,(で)こと，古村氏に製品概要レポートしてもらいましょう。

私がX68000に転んだ理由

本題に入る前に私の個人的な話など，ちょっと聞いていただきましょう。知っている人もいるかもしれませんが，私はFMユーザーでした。いえ，今でもFMユーザーで，私は下宿ではX68000，実家に帰るとFM-7を使っています。実をいうと，このX68000も去年買ったものでそれまでの4年間はずーっとFMオンリーのユーザーだったわけです。では，なぜ私がX68000ユーザーになったか？　ほとんどのFMユーザーと同様に私も富士通から68000マシンが出ることを待ち焦がれていたのです（でも日経パソコンによると期待の32ビット機 FM-TOWNSは386マシンだそうです)。

私はFM-7では主にF-BASICのエディタを使うアブソリュートアセンブラを使っていましたが，そのほかにもOS-9のBASIC09（8ビット版のOS-9用のBASICでプロシージャなどが使えるキッカイなものでした）などでよく遊んでいました。当然のようにメモリも64Kバイトしかなく，ディスクも遅くてうるさい5インチ2Dという劣悪な環境ですから，コマンドを実行するたびに（OSのコマンドであるLISTやDIR文でも）ギカギカと音をたてながらチンタラとディスクを読みにいくわ，それがいやで主なコマンドをload文でメモリ内に置いておくとあっというまにメモリが足りなくなるわ，エラーメッセージを表示させるのにprinterrというコマンドが必要だわで，とても悲惨な状況でした。また，某誌で酷評されたようにCP/MやMS-DOSなんかに比べるとアプリケーションはない，マクロアセンブラもない，エディタもたいしたものがなかったのも事実です。

でも，OS-9/6809には夢がありました。マルチタスク，マルチウィンドウとワークステーションクラスのマシンでしかできないような芸当を8ビットパソコンがなんとかやってのけるのです。そして，そうした環境があればなんの苦労もなしに（あったけど)，実行しながらのデバッギングなどという未来のパソコンにしかできないようなことがあのチャチなFM-7上で気分だけでも味わえたのです。

そう，MS-DOSとOS-9シリーズのいちばんの大きな違いっていうのは，まさに現実を追求するか夢を追うかの違いだと思います。とりあえず，アプリケーションを載せて絶対番地で速く動かし，I/Oを直接いじることもできるMS-DOS。かたや，少しでもミニコンやワークステーションなどパソコンの理想への道を求め，多少きつくても，TSS機能やマルチウィンドウなどの機能を詰め込めるだけ詰め込んだOS-9。そして，私も理想を求めるOS-9に感動し，もう一度OS-9を使ってみよう，次のマシンもOS-9/68000が動くマシンにしよう，そう考えたのです。

よって，「X68000にOS-9が載る」，その話を聞いたときから，私の心はX68000派になっていました。富士通はMS-DOSでいっている。かたや，X68000にはOS-9が載る。私は私の夢を求める決意を固めFMの新機種を待つのをやめ，X68000ユーザーになったのです。

OS-9，一丁，お待ちどう!

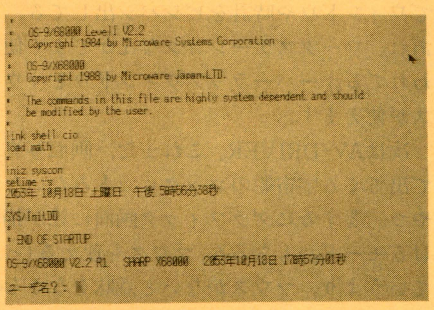
そして，ついにX68000用OS-9/68000が発売されるときがきたんです。

(編)「OS-9のディスクとマニュアル。よろしくね」

(で)「こ，これが……（嬉しくって言葉にならない)」

というわけで，終電で帰ってきたにもかかわらず，私はすぐにOS-9でひと晩中になってしまったわけなんだな，これが。なにしろ，X68000を買った理由だったものにやっと会えたんだから。

さてっと。なんか知らないんですけど(っていうか説明は受けたはずなんだけど，舞い上がってたせいか，なーんにも覚えてないんだな，これが)。OS-9のディスクがいっぱい。えーと，どれがどれなんだ？　ま，



これが起動画面だ!

とりあえずシステムディスクを入れてと。お，お，お，これはっ！　げげ，本気でオーバーラップウィンドウしてる。おまけにユーザー名まで聞いてくる。ほとんど，ワークステーションだなー，こりゃ。

あ，すいません。ひとりで遊んでて。えーっと，製品版の概要を解説しておきましょう。OS-9のディスクっていうのは，4枚組になっていてそれぞれ「システムディスク」，「AV-RIDERプログラム」，「AV-DRIVERデータディスク」，「ASK68K辞書ディスク」という名前がついてます。

で，システムディスクにはOS-9のコマンドやエラーメッセージなどが，AV-DRIVER用の2枚にはAV-DRIVERのコマンドとデモのデータ，そして辞書ディスクにはHumanでお馴染みの日本語フロントエンドプロセッサASK68Kの辞書が入ってます。

標準でついてくるコマンド数は83とHumanの48に比べ倍増してます。マニュアルもほぼ倍の厚さで，用語集までついているんでOS-9は初めてという人(たいていそーだろーな）にもいいんじゃないでしょうか。

あらかじめいっときますけど，シャープから発売されるのは本当にOS本体とまわりのこまごましたものだけで，BASIC09やC，アセンブラなどはついてきません。エディタもラインエディタだけ。ほかのおいしそうなものは御本家のマイクロウェアから，出てくるそうで，とりあえずは「C＆プロフェッショナルパッケージ」を揃えておきたいところですね。これがないとなーんもできません。肝心のお値段は，OS-9本体が29,800円，Cが58,000円とのこと。

まずセットアップ

で，AVなんとかだ，システムだっていうのがややこしいですが，これはX68000のOS-9 にはマンマシンインタフェイス（平ったくいえば，画面モードとマウスなどをどう使うか）に３つのモードがあるんです。

まず，最初がパーソナルウィンドウ。これはよくOS-9の画面写真に出てくるマウスでウィンドウや時計をビシバシ出してる，あれ。ワークステーションなんかでよく使われてるオーバーラップウィンドウとマウスが使えます。

次はAV-DRIVER。これまた，画面写真で出てくる宇宙船のコックピットみたいなやつ。要するにグラフィック画面に絵が描けるモード。もちろん，マウスも使える（っていうより，マウスがないと意味をなさない）。これがX68000を使ってるっていちばん実感できるモードかもしれない。

んでもって，最後がマルチスクリーンディスプレイモード。昔っからある，「OS」って感じで画面は黒地に白い（ウィンドウによっては黄色とか青も）文字だけで構成されてる，Human68kのCOMMAND.Xに近い感じで，もちろんマウスなんか使えない。でも表示される画面の裏に，あと３つウィンドウがあって４つちゃーんとマルチタスクしてる。ちなみにウィンドウの表示ページを変えるには[OPT.1]＋[ウィンドウのナンバー]です（あ，OS-9を知ってる人に注意！　マルチタスクにしようとして，わざわざ，

$shell </term2>/term2>>/term2&

なんてやらないように！　ちゃんと初めっからなってるんだから）。

ま，説明が長くなっちゃったけど，そういう，３つのモードが入ってるわけです。で，どれかのモードが動くディスクをmake_d0_??? とかやってそのモードのバックアップディスクを作ればディスクを立ち上げたとたんにそのモードになるディスクを作ってくれるようです。

ハードディスクを使っている場合には，一度バックアップを取って OS-9 でフォーマットしなおすようにします。たとえば，20MバイトをHumanとOS-9で10Mバイトずつ使いたいとかいう人もいるんじゃないでしょうか。そういう人も大丈夫です（でも，どっちか 9M バイトになっちゃうんですよね）。それ用のツールを使ってやれば，起動時にメニューが出てHumanにするか，OS-9にするか聞いてくるのであとは選んでやるだけです。

これで実際にOS-9が使えるようになったわけです。あとは12月号の桒野さんの記事でも見て，皆さん，パーソナルウィンドウでびしばししちゃってください。

Humanとのデータ互換

さて，バックアップのディスクができたところで今回めでたく発売となった OS-9でX68000専用につけられた機能を見てみましょう。

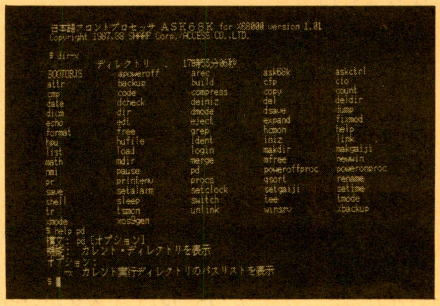
まず，X68000には標準で Human68k というMS-DOSライクなDOSがついてくるわけですが，ふつうに考えたらOS-9と Humanとはフロッピーディスクの物理フォーマットから違いますので，OS-9でHumanのデータを読もう（逆でもいいけど）などと考えたらファイルコンバータからなにから自分の手で作らにゃならんわけで，そりゃまた大騒ぎになってしまいます。

ま，そんな間抜けな事態は X68000 では考えられないわけでちゃーんと「HuFILE」なるモジュールがついてきて，しっかりとHumanのおいしいところはめいっぱい持ってこられちゃうんだよねー，当然。

で，このコマンドは実に多くのサブコマンドを持っていて機能も非常に豊富です。軽く見ていくとHumanのディスクに対して行うCD, COPY, DEL, MKDIR, REN, RMDIR（みんな Human 上のときと同じ意味のサブコマンドです），Human と OS-9でデータの転送を行う GET，PUT，挙句の果てにTYPEやDIRもあります（もちろんHumanのテキストファイルをのぞき見たり（OS-9でいうLIST），ディレクトリを取ったりするんです）。

実際，このコマンドをいちばん多く使うだろうと作る側も考えたのか，マニュアルではコマンドの説明としてはHuFILEにもっとも多くページ数をさいています。

また次に，いま現在でOS-9（もちろんOS-9/68000ですが）が動くパソコンっていうのはFM-11/16β，PC-9801に 68000ボードを搭載したものぐらいだと思うんですが，どっちみちスプライトだFM音源だと豊富な機能を搭載したマシンでOS-9が動くのはたぶんX68000しかないでしょう。で，それ用のモジュールがあります，やっぱり。



マルチスクリーンシェル

それはAVPLAY。これひとつだけです，AD PCMもFM音源もグラフィックも。えーっなんでぇ？　とおっしゃる方もいることでしょう。さーてなんででしょう。当たったらじゅってんあげよう，なーんてね。

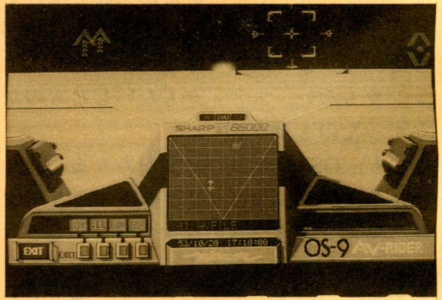
実をいうとこれはさっきのHuFILEともの凄く関係があります。まず，AV 関係でAVPLAYとHuFILE以外のコマンドは ARECとDMODEがある。ARECというのはAD PCM のデータをサンプリングしてファイルに落とすコマンド。DMODEは画面の解像度を256×256から768×512まで各種変更するモジュール。じゃそれ以外のデータをどうするかというと，実はHuFILEコマンドでHuman68kから転送してきたデータを使うのです。まず，

1)　X-BASICのIMG関数
2)　Z's STAFF PRO-68Kで作ったデータファイル
3)　MUSIC PRO-68Kで作った音楽ファイル
3)　SOUND PRO-68Kで作った音色データ
5)　ユーザー定義のパレットデータ

のいずれかをHuFILEのGETサブコマンドでOS-9のファイルにコンバートします。そして，それをAPLAY に通せば，できあがりっ！

そうそう，HuFILEにはあと，COMPRESS/EXPAND（IMGデータの圧縮，展開です）なんていうのもあります。まあ，MUSICデータ（拡張子が .mml のやつ）なんか，見てみるとHumanの .opmファイル（つまりHumanのMML）とほとんど同じなのでわざわざHumanなんかで作って持ってこなくてもスクリーンエディタがあれば自分で書けそうな気がするんだけどねー。

それから，前にテレビを見るときも Human68k と同じように[OPT2]＋[チャンネ



AVシェル

ル番号］です（そりゃそうだわ）。

どうも，このOS-9もかなりHumanを意識して作ってあるみたいです。ふつうの別売りのOS（DOS）ていうのはまっとうなファイルコンバータってほとんどなかったから，大きな進歩といえるでしょう。たとえば，X1のCP/MではついてくるのはCP/Mで組んだプログラムをBASICのフォーマットの「カセットテープ」（テープですよ。ディスクじゃなくて）とのコンバータだったし，PC-9801のMS-DOSとBASIC間にもコンバータはついてきた記憶はないし，FMでもOS-9, F-BASIC間は確か自作のツールでコンバートせざるをえなかったように思います。

もっとも，あれだけまともなコンバータがあると雑誌に「ファイルの転送ツールを作りました」なんていうお手軽な投稿ネタが使えなくなってしまうのはさみしい気がしないでもないですけど。

そしてAV-RIDER

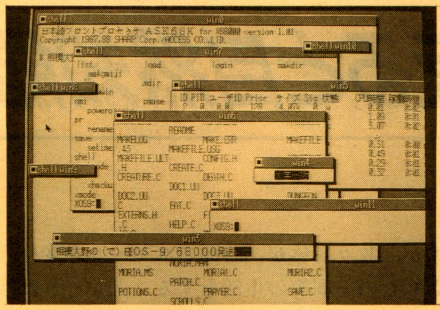
でもって，FM音源や，グラフィックのデータをマウスだけで再生する方法もあります。いわずと知れたあのAV-RIDERのデモで出てくる宇宙船のコックピットに似たビジュアルシェル(!?)です。要するに，前方から飛んでくるファイルアイコン(？)をマウスで撃ち落とすとAVプログラムが起動するという，なんていったらいいか……とっても興味深いシェルなんです。こんなものをOSのシステムに入れるほうも入れるほうなら，このマニュアルも変なノリです。

「ブートアップのメッセージが表示されたあと，宇宙戦闘機のコックピットのグラフィックが表示されます。この戦闘機はAV-RIDER68タイプ１と呼ばれるもので本機のニックネームはジョナ（JONAH）です」（マニュアルより抜粋）。

しょーがないなー。きっと，OS-9の開発チーム（あるいはマニュアル書いちゃう組）にはきっとアニメマニアがいるに違いない。もっとも私はこういうノリは好きだけど（うー，いかん，だから，オタッキーなどと呼ばれてしまうんだ）。

辞書なんかどう？

そうそう，このOSには日本語フロントエンドプロセッサとしてASK68Kがついてくる。どんな86マシンのMS-DOSにもフロントエンドはついてくるし，X68000のHuman68kにもASKがついてきた。だから，当たり前といえば当たり前なのかもしれないけど，OS-9では標準で漢字入出力用のドライバがついたのはたぶんX68000が初めてだと思う。だいたいなー，FM-7のOS-9/6809じゃなー，漢字がANK文字の縦2倍で表示されちゃうからボコボコな表示になっちゃうらしいんだぞー（実際に使ったわけじゃないから詳しくはわからないけど）。



日本語処理もこのとおり

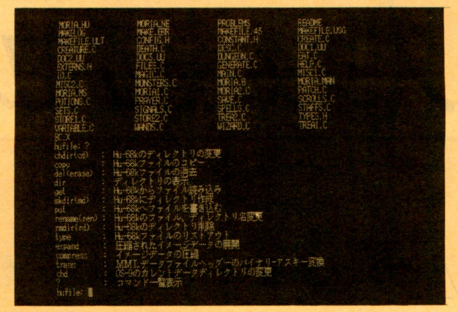
ま，それでだ。Humanのフロントエンドと，おんなじ名前のフロントエンドが載ってる。いや，名前だけじゃなく操作法からなにからほとんどおんなじ。そう，実はHumanのフロントエンドとは使い心地は瓜ふたつ。で，なんと辞書も共通なのでさっきのHuFILEコマンドでHumanのASKとOS-9のASKで辞書のやり取りができるのだ。しまったーっ！　私はずーっとフロントエンドにはe1.sysを使っているのでOS-9では使えないのだ。むーん，くやしい。

私みたいなお馬鹿はほっといて（わー，見捨てないでー！）あと，日本語関係としたら，DICMという辞書の保守管理ユーティリティとMAKGAIJI, SETGAIJIという２つの外字関係のコマンドがあります。これも例によってHumanの外字ユーティリティUSKCGMでできた外字ファイル「USKCG. SYS」をHuFILEでOS-9に持ってきて外字をセットするものだったりします。

そうそう，キーボードもね

ところで，いっくらウィンドウが開けて，マウスでビシバシとポインティングデバイスできてもやっぱりTK-80BSやLKIT8の時代からパソコンとキーボードは切っても切れない仲になってしまっています。

ところが，このOS-9というのは初期設定の時点ではコントロールキーと「？」（？はキーボードの１文字です）の機能が違ってるんです。たとえばですね，「この前打ったコマンドをもう１回打つのはだるいなー」というときに打つキーはHuman68k（っていうかMS-DOSっていうか）では［F3］キー（または［↓］キー）なわけですよね。



Humanのデータを見る

これがOS-9では［ctrl］＋［A］だったりするわけです。さーて，ではなーんにも打ってない状態，つまりユーザー名をOS-9が聞きにきたときに［ctrl］＋［A］を打つといったいどうなるんでしょう？（自己顕示欲でしょうねえ）

あ，話がそれてしまいましたが，［ctrl］＋［？］はTMODEというコマンドで対応する機能を（BSとか，→とか）いろいろ変えられますので，ぜひ書き換えてしまいましょう。

あなたも一緒にハマってみない？

オセッカイかもしれませんが勝手に私がX68000用OS-9/68000についてまとめてしまいたいと思います。

まず，OS-9というのはマルチウィンドウ，マルチタスクを特徴とするOSで，当然そのどちらもX68000の性能を生かせられるように，特にマンマシンインタフェイスについてはかなりしっかりと作られたOSである。

そして，X68000の特徴である，FM音源やAD PCMについてもAV-DRIVERのディスクのなかにすべてしっかりと詰まっていて，そのデータはすべてHuman上の財産を継承できる。EJECTなどのコマンドも完備している。

また，漢字変換に関してはHumanと同じASK68Kを日本語フロントエンドプロセッサに採用しているので操作法などでとまどう心配もなく，またHuman上で鍛えてきた辞書も十二分に活用できる。

つまり，データはHuman68kからぱっくんできるバリバリの新OSといえると思います。

＊　＊　＊

OS-9ははるか先の理想のパソコンに近い思想のもとで作られたOSです。しかし，それが本当に理想のものとなるか，あるいは誰も見向きもしない塵同然のものになっていくのか，それはこれから私たちが決めていくのだと思います。だから，私はいいます。「あなたもハマってみませんか？」

人工知能が肥えると自然知能がやせ細る

どうしても超えられないもの

いつの時代にも，うっ！ と驚くような進展(発明，発見，開発など)に少なからず遭遇しますから，その積み重ねによる現代文明の成果は実に大きなものでしょう。特に，電子計算機なるものは，作られてほんの少ししかたっていませんから，その前途はまだ未知数です。したがって，そこにおける人間のありようなどは，まったくといっていいほどわからないでしょう。

とはいえ，人間がどうしても突破できない制約というものを挙げるとしたら，次の3つだと思います。

1) 空間的制約

世界は3次元の空間でできており，遠い，近いという尺度が存在する。それにより，物体の移動などに伴う制約が生じる。

2) 時間的制約

時間は一方向へ連続的に単調に進行する。そして生物の命は有限である。

3) 物質的制約

世界には固有の性質を持った物体が存在しており，そこには本質的に付随する制約が生じている。

突き詰めればこのどれかに相当するようなさまざまな制約を，なんとか打ち破ろうとして，人々はあくなき挑戦を繰り返してきました。その結果，人は地球上をかなり自由に（地球の外にも飛んで行きそうな勢いで）移動し，物体をかなり好きなように利用できるようになったわけです。

つまり，コストの点を除けば，大昔に比べて空間的・物質的制約はそこそこ除外できてきたといえるでしょう。

してみると当然，残る時間的制約が，目の前に大きな障壁となって立ち塞がるのでありました。そして，「ああ，1年でも多く長生きしたい！」と考えるわけです。

近寄ってくる情報たち

どんな生物に対しても，同じようにまた一様に，時間は過ぎていくように見えます。この時間的制約から，人はどうしても抜け出せないのでしょうか？

時間的制約を直接取り払うことは，それこそタイムマシンでも想定しないと無理ですが，我々自体が，時間のスケールを変化させたのと同等の生活をすることができれば，相対的に時間の制約を解消することにつながります。

つまり，日常のさまざまな行動に要する時間を短縮でき，しかも短縮前と実質的にはまったく変化がないようにできれば，その短縮率の分だけ時間的制約を取り除けたことになります。

こうしたことは，近代・現代における合理化の概念が目指す方向ですから，あらためて説明するまでもないこととは思いますが。

ところで，人間をブラックボックスとして捕らえ，入力と出力のみ捕らえる立場があります。そして人間の知的能力，情報処理能力自体を大幅に向上させることが不可能な場合，このブラックボックス全体の処理速度を上げるひとつの方法として，できるだけ人間という情報処理装置が処理しやすい形に入力を変換する，というものが考えられます。

そうしたことは，すでにいろいろな社会的現象として見られます。具体的な表れとして，漫画の大流行（少年ジャンプはなんとついに500万部！）や，ちょっと古いけど写真週刊誌（FF現象とか呼ばれてた）などがあります。それは，2次元の絵や写真から得られる情報量あるいは情報獲得速度が，1次元の文字列から得られるそれに比べるまでもないほど大きいからといえます。

このように，情報を人がなるべく速く獲得できるような形式に変えることは，「情報が人間に接近してきた」と換言できるでしょう。茶の間でテレビをつけると世界中の様子がすぐそこに見える(ような気がする)ことなどは，如実にそれを表しています。

情報が我々に接近し続けると，最後には我々の脳の神経細胞のパターンそのものが，情報の表現・記録手段になるという事態に至ってしまいます。このようなことをテーマにした「ブレインストーム」という怖い映画があります。脳の中の情報を自由に記録・再生する操作が引き起こす現象について描かれていました。いまだに解明されていない部分の多い脳に対して通常でない刺激を与えることの影響がたいへん不気味だったことを記憶しています。

これはSFの世界にとどまる話題ではありません。学問的にも，そうした脳への直接的な入力に関するテーマを取り扱った発表を一度，目にしたことがあります。コピーしてとっておいたのですが，文献類の山の中に音もなく（当たり前か）消えてしまいました。申し訳ありませんが，いずれ見つけたらご紹介します。

人工知能に頼る自然知能

入力と出力を持ち人間という名がついている情報処理装置で，装置自体を変更せずに処理速度を上げる方法のひとつとして，入力する情報を人間に近づけていく，というアプローチを考えたわけですが，次に考えられるのは，入力情報のうち簡単に処理できてしまうことは，人間の脳を通さずに付加装置で処理してしまう，というやり方です。これは全体のスループットを向上させるうえで当然大きな意味があるといえます。

世界では，もうすでに多くの処理はマシン，特に電子計算機が行うようになっています(株の売買だって自動的にやっている)。そして，この連載で終始横道にそれながらもじっと斜めから見据えているつもりの「人工知能」あるいは「知能機械」自体こそ，未来においてそのような意味づけを持ってくるものと思われます。

先ほど指摘した「空間的制約」に関していえば，自動車や飛行機などの発明に伴って，生活でも身体的な制約が緩くなってきたため，人の身体は退化を続けてきました（ベン・ジョンソンすれば別ですが）。

同じように，頭を使わなければできない仕事についても，それを知能機械に任せて自分ではやらないとなったら，当然その能力は退化することでしょう。それはどうしても避けられない厳然たる事実であると思います。したがって，何を機械に任せて何を任せないかという判断が重要になってくるわけです。

知能機械というとまだ先の話のように思われるかもしれません。しかし，たとえばワープロが普及した結果，人は漢字を読むことはできても書くことに大きな抵抗を感

じるようになったり，書き順という概念の意味がなくなったりすることも想定できます。電卓の使い方はうまくなって自在に高度な計算ができても，暗算はできなくなるかもしれません。

ここが見方の違いであって，結局いろいろな計算もできるようになるのだと考えるか，暗算すら自分ではできなくなるのだと考えるか，差が出てくるのです。

そのうち出現するであろう（見方によってはすでにその一部は姿を見せつつある）知能機械を，人間の知能の拡張として捕らえるような楽観的な見方をすることは，知能機械の研究開発を促すという点で必要なことはもちろんです。が，同時に知能機械を取り去ったあとの人間の知能自体の姿も，絶えず想定していないとまずいんじゃないかなぁと僕は思っています。

人工知能と共存する社会

先日，あるニュース番組を見ていたら，コンピュータウイルスのことを取り上げていました。被害にあった人の証言も含めて珍しく具体的にレポートしていたのですが，不謹慎かもしれませんがつい笑ってしまいました。というのも，被害の状況を示すためにいろいろ計算機のモニタ画面を見せてくれるのですが，どうも僕の理解ではそこに出ていた，

delete ＊.＊

というバッチファイルの中に書かれていたコマンドが，単にその人の大事なファイルを消してしまっただけのように思えたからです。

なにか潜在するメカニズムによって，次々と他人のファイルを消していくならば話は別ですが，どうもその特定の人に対する嫌がらせのためにそのコマンドをこっそり（起動時か終了時の）バッチファイルに書き込んだだけのように見えたのです。そうだとしたら感染力はないわけですから，ウイルスとはいえませんね。

感染するにせよしないにせよ，これはやはり犯罪として取り締まらねばならないでしょう。計算機を始終使っている人にとっては，あるいは真にパーソナルな計算機（つまり知能機械）を得ているであろう未来の人にとっては，自分の脳を拡張したものとしての知能を傷つけられることは，身体や頭脳に損害を受けるのと同様のダメージになるのですから。

このように，計算機にまつわる犯罪や悲しい話は，要するに知能機械に依存する程度が深まったということにその源があるといえます。その依存度が高まっていくと，つまり理想的な知能機械が出現したときには，過激な言い方をすれば，それは人間を超えた存在になっている，と主張する人がいてもおかしくありません。この連載でもご紹介したとおり，「進化の次の段階は人工知能である」（アメーバからずっと進化して猿，そして人間，次に人工知能という進化をいっている）という学者さえいるのです。

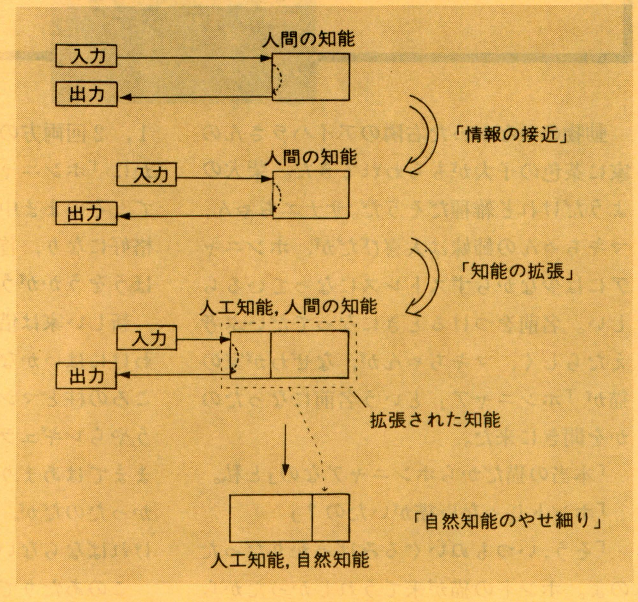
人工知能と共存する社会，もっといえば人工知能が（人間の能力ではなかなか難しい）グローバルな制御を行う社会における力関係，権力の所在というテーマもまた難しい問題です。結局は知能機械をうまく操作できる人が権力を握ることもあり得るかもしれません。

あるいはそこにこそ現在いろいろな面で研究の場が拡大してきた「分散処理」（コネクショニズム，分散オペレーティングシステム，分散アルゴリズム，ネットワークなどなど）が必要とされるのかもしれません。力の一点集中を避けるために知能機械自体を分散することはストレートな解である気がします。

人類のアイデンティティ

人類は，その出現からこれまでの長い間に何度かの革命的な出来事に遭遇してきました。太古の時代から使い出した「道具」というもの，そして科学の行きつく先としての「人工知能」は次の革新を構成することでしょう。

図　情報処理装置としての脳およびそのやせ細り



そして人工知能が次から次へと人の知的な処理をうまくこなせるようになってくるときになって問われるのが，人間としてのアイデンティティ（自己同一性とか存在意義）だと思います。最初は，数値計算などの機械的な処理だけを任せていたのが，次第に知的な情報処理までも人工知能に任せるようになっていった場合，結局なにが人間に残されるであろうかということなのです。たとえば，その答えとして「創造的なこと，たとえば芸術や文化」ということがまず挙げられるかもしれません。でも，こういう発言をすると怒られるかもしれませんが，いま流行っているような芸術や文化と呼ばれるもののうち，知能機械でも（将来においては）生産可能なもの（つまりワンパターン的なもの）を取り去ったら，どのくらいあとに残るのだろうかなどと考えると，これまた，わけがわからなくなってしまいます。

やはり，入力と出力だけで考えるブラックボックス的な情報処理装置として人間を考えた出発点からこれは間違っているのでしょうか。

どうも僕は，人工知能自体の実現に関してはかなり楽観的であり，一方そのときの人間の存在という点についてはまだまだどうなっているといった感じのようです。

第32回

猫とコンピュータ
猫にマウス？

Takazawa Kyoko
高沢 恭子

さて，待望のニューマシンが到着した高沢家はさらに活気づいているようです。え？そうですか，でも恭子さん，一番初めは誰だって何かつまずいたりしますよ。そこがまた楽しいじゃありませんか，ね？

動物のいなかった右隣のアイハラさんの家に茶色の子犬がもらわれてきた。柴犬のようだけれど雑種だそうだ。サナエちゃん，マキちゃんの姉妹は大喜びだが，ホンニャアには少なからずストレスになっているらしい。名前をつけるときになってだいぶ考えたらしく，マキちゃんが，なぜわが家の猫が「ホンニャア」という名前になったのかを聞きに来た。

「本当の猫だからホンニャアなの」と私。

「ホントじゃない猫がいたの？」

「そう，いつもぬいぐるみばっかりだったのよ。ホントの猫が来てうれしかったからホンニャアになったの」

「ウチもホントの犬が来てうれしいけど，ホンワンじゃヘンよネ」

そのあと決まった名前はハチ公物語にちなんだのか「ハチ」になったそうだ。

ホンニャアが庭に出ると，右隣の庭で太った可愛いハチがせわしなく動きまわっている。左のソノダさんにはマルチーズのモコがいるし，なんだか犬にハサミうちされたかたちのホンニャアのこのごろなのだ。

ストレスは材木で

ホンニャアがこのところ部屋の柱をバリバリとひっかくのは，ただのツメとぎではなくてそんなストレスを発散させているのだろうか。

S市の庭は十分な広さがあったので，あちらこちらにホンニャアのツメとぎ用の自然木が置いてあった。フジ棚の下や花壇と庭石の間など，彼は好きなところでセッセとバリバリやったものだ。

これらは，部屋の中の柱やタタミをホンニャアのツメの被害から守るためのものだったけれど，それでもやっぱり室内のそこかしこでよくバリバリとやりかけた。

気に入りの柱につかまって立ち上がり，1，2回両方の前足が行き交うと，必ず誰かに「ホンニャア！」と声をかけられるので，そのまま中腰でストップモーションの格好になり，首をうしろに反らせ呼ばれたほうをうかがうポーズに愛敬があった。

新しい家は借りものだから，傷をつけるわけにはいかない。食卓にいちばん近いところの柱とマシンルームの入口の柱が，どうやらレギュラーポジションのようだ。いままではあまりツメとぎ行為は目にたたなかったのだが，これは早急に対策をたてなければならない。

このあたりで適当な木材を手に入れるのはむずかしいと思われたが，例の双子の三毛猫のいる商店街の酒屋さんでちょうだいすることができた。

店の出入口のわきに捨てられるように置いてあった1mくらいの角材を，いつも店番をしているおばあちゃんに「もらってはいけないでしょうか」とたずねてみた。

80歳近いおばあちゃんはじっと私を見ていてなかなか答えてくれない。ダメかしらと思っていたら，店の裏のほうを指さして，

「まだ，ほかにもあるよ」と言って立ち上がりそうにした。

「いいえ，1本あればいいんです。猫のツメとぎ用に庭に置こうと思って」

「猫がいるんかい」おばあちゃんは座りなおして聞いた。

「ええ，柱でツメをとぐもんで叱るんですけど，どこにもツメをとぐところがなけりゃかわいそうですから」

「ウチの猫はそこらの段ボールをみんなひっかいてるよ」

「あ，あの2匹の猫ちゃん，ウチの庭によく遊びにきますよ。団地の向かいですけど」

「あれまぁ，どこまで出かけてるかちっとも知らないからねぇ」

大柄で少しいかめしい顔のこのおばあちゃんが笑ったのを，私は初めて見た。

角材を自転車に乗せて家に帰り，さっそく庭のまんなかに置いた。少し汚れているけれど，どうせそのうちホンニャアがツメで削りとってきれいにしてしまうだろう。もう1本くらいいただいてきてもよかったかもしれないのだが，この小さな庭に角材2本は目ざわりすぎる。

あとはホンニャアが部屋の中でバリバリとやりはじめたときに，丸ごとつかみあげて庭のこの角材の上にのっけてやればよいのだ。その点，彼はあっさりと思いきりの良い猫なので，無駄な抵抗はせずにひきつづきその場でバリバリとやるのがいままでの例だ。彼は一度でここをツメとぎの場所だと認識してくれる。ただし，だからといって決して部屋の中でやらなくなるわけでもないというのも，いままでの実績である。

灰色の黒船

庭にツメとぎのツールができてまもなく待望のX68000が到着した。

いかなる思いつきであれ，自分のためのコンピュータを買い入れたのだ。こんな日がくるなんて誰が想像したことだろう。狛江のアニキが知ったらなんて言うかしら。

「おお，ついに類人猿も自分のマシンを持つ日がやってきたかぁ……感無量だ」

たぶんそんなことを言うだろう。

夫といっしょに何軒かの店を下見して決めたのはX68000 ACE HDという機種である。黒とグレイの2色のうち，指定したグレイがその日店内になかったので，届けられたときが初めての対面だった。

包みを開く。真新しい肌ざわりへのときめき。でも予想していたよりも暗めの灰色だったせいかそれが重厚さと共にとてつもないパワーを思わせて，一瞬よぎった後悔は決して小さいものではなかった。

だけどやっぱり嬉しい。新しいことを手がけようとする喜びと期待があふれてくるようだ。そうだ，こんどはマニュアルを真剣に読みすすめながらやっていこう。

いままで何台パソコンがやってきても，一度たりともきちんとマニュアルに取り組んだことなどない。でも，こんどは違う。意欲も心がけも違う。目指すZ'sSTAFFは準備を整えてから購入しよう。

ハードディスクに転送する

本体とディスプレイはマシンルームの一番右の奥に置くことになった。

棚に載せるには夫の手を借りたが，ケーブルやコードはマニュアルを見て自分でつないでみた。一人前になったようでなかなか良い気分だったが，TV CONTROLとある場所にコネクタが見あたらなかったときはあわてた。が，すぐに保護色のキャップがかぶせてあるのに気づいて笑ってしまった。やってみるのが一番の学習だ。

パソコン誌などをひもといてみると，マウスを使った操作はもうすっかり定着しているようだが，私はほとんど経験がない。X68000でも，このマウスを駆使することで，キーボードが主流だったいままでのパソコンの操作を，たやすくしかも幅広くしたことが特徴となっているらしい。

キーボードをいくらか叩きなれてマニアモドキを気どっていたのに，シンボルマークをデザインしたアイコンにマウスを這いまわらせるなんて，なんだか逆戻りしたようで不満な思いもある。確かに，操作をシンプルにできるのは作る側の実力だろうけど。でもキーボードのほうがすばやく特定できる場合だって多いんじゃないかしら。

ともあれ「親しみやすい高性能マシン」がキャッチフレーズのX68000だそうだ。「寛大な知恵者」に見放されないように，私もがんばらなくてはいけない。

さて，ご承知のようにこのマシンには20Mバイトのハードディスク1基が内蔵されている。まず，この内蔵ハードディスクにシステムや辞書ディスクの内容を転送して，ハードディスクからシステム起動できるようにしなくては。せっかく，フロッピーディスクに比べてはるかに大量のデータを高速に扱えるというハードディスクタイプを購入したのだから。

まずハードディスクの領域確保をする。マウスの使用から，途中コマンドモードに切り換わり，指示に従って操作を続ける。フォーマットが終了すると再びフロッピーディスクからの起動を行い，ディスクの内容をすべてハードディスクに転送する。

やった！　指示どおりぬかりなく。これからはハードディスクからシステム起動ができるというわけだ。

やっぱりゲームだ！

「パーソナルワークステーション」X68000を待ちわびていたのは私だけではなかった。注文もしないうちから届くのはきょうか明日かと問い続けてその日の到来を心待ちにしていたのは，もちろんトオルである。

その性能や専用ディスプレイテレビの高画質，絵も描きたいしサウンドソフトへの期待もある。そしてなにはともあれ小手調べにゲームをやってみたい。

ゲームはかねてよりX68000版の「源平討魔伝」が強く心にかかっていたので，マシン到着にあわせてそれとなくパパにねだっていた。ところが「源平」の品がすぐにはないというので，「沙羅曼蛇」のほうをとりあえず買ってもらうことになった。まあ画質の鑑賞と諸々の研究をかねて，ゲームがひとつ2つあるのも悪くはないし，あの陽気なゲームという出し物があってこそパソコンは花が咲く。

マシンと日をたがわずに到着した「沙羅曼蛇」は，私がマニュアルを必死に読む横で，さっさとトオルにトライされていた。

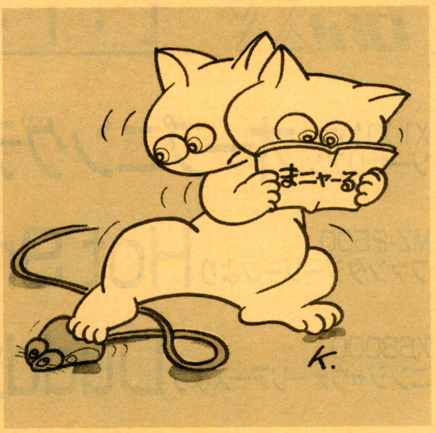
彼は新しいマシンであってもほとんど何の躊躇もなく，しかも誤りもせずに操作して，ハイビジョンを思わせる画像の中で「沙羅曼蛇」を縦横に楽しんでいる。

さて，ここで私が内蔵ハードディスクへの転送をしてから異変が起こった。

トオルがゲームのディスクをフロッピーディスクドライブに挿入したが，まったくスタートできなくなったのだ。

「なぜェ，どうしてかなぁ」とトオルは挿入やリセットを何度もやりなおしてみるが変わらない。どうしてかといえば，ハードディスクに転送したときから事態が変化したとしか考えられないではないか。

マシンと，だんだんキゲンが悪くなるトオルとをいっぺんに修復するには，ハード



ディスクに転送した内容を撤去して，元の状態に戻してみるしかない。さて。

元の状態に戻すにはFORMATの手順を行い「処理の選択」で「装置初期化」を選べばよいのではないかと考えた。そこで「装置初期化」を選択したら，「何メガバイト確保しますか？」ときたので「1メガ」とやると再び同じことを聞いてきた。これは困った。マニュアルをあちこち探すと，「もし誤って装置初期化を選択した場合は^C(CTRL+C)で抜けてください」。そこでしめたとばかりにCTRL+Cをやり，これですべてが解決したかに思えた。

ところがこれ以降，何のディスクを差し込もうと，「領域確保されていません」のメッセージばかりで何も起動しなくなった。ああ，フロッピーディスクから立ち上げることもできなくなってしまったとは。

あのォ　モシモシ……

トオルの失望は大きかった。私の反省は深かった。かくなるうえは，メーカーの「消費者相談室」に問い合わせるしかない。北区にある東京支店の消費者相談室へ電話してみる。

ていねいな応対をしてくれた。「確かマニュアルにあるはずなんですが……」

ほんとうに「取扱説明書」に記してあった。あわて者の目には肝心なものが映らないらしい。「OPT.1(オプション1)キーを押しながらリセットスイッチを押してください。すると，フロッピーディスクから再起動します」

これでめでたく「沙羅曼蛇」はよみがえった。デタラメは禁物，1にマニュアル，2にマニュアル。だけどハードディスクのほうはどうなったのか，まだ未解決だ。

X1/X1turbo
ソーサリアン **オープニングテーマ** ©日本ファルコム
Nishikawa ZenJi
西川 善司

MZ-2500
ファンタジーゾーンより **Hot Snow** ©SEGA
Yoshida Takenori
吉田 健智

X68000
ニンジャウォーリアーズより **Duddy Mulk** ©TAITO CORP.
Doi Atsushi
土井 淳史

今月は3本です。作品を送ってきてくれる人たちも，ゲームミュージックファン，ロック/ポップスファン,クラシックファンと分かれるようですね。今年はいろいろ挑戦してみると楽しいですよ。サウンド関係にはまだ手を染めてない，という人もぜひ参加してください。

MusicBASICしてますか

さて，ゲームファンの皆さん，お待たせしました。今月は，久し振りにゲームミュージックの作品ばかり，続けて3曲お届けします。

まずはOh! X 1988年12月号で発表されたMusicBASICのサンプル曲から始めましょう。ソーサリアンのオープニングテーマです。コンパクトにまとまっていますので，ぜひ入力してください。作者は，MusicBASICを作ってくれた西川氏。12月号にはやはり彼の作になるソーサリアンのエンディングテーマも掲載されています。

曲を聞くときは，MusicBASICを起動したあとリスト1のプログラムをrunしてください。演奏がスタートします。

さて，皆さんはもうMusicBASICは入力し終わりましたか。投稿も届き始めました。いずれ大々的に発表したいと思いますので，どしどし力作を送ってください。期待しています。



ニンジャウォーリアーズ

ファンタジーゾーンです

MZ-2500用には，SEGAのファンタジーゾーンからの1曲です。

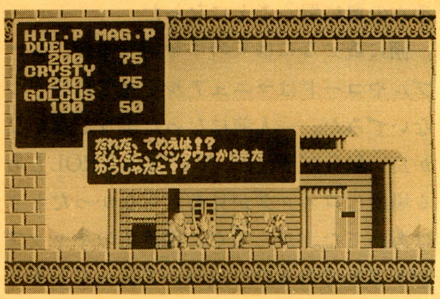
横スクロール型戦闘ゲームにRPGの要素が加わって，2年前にビデオゲームとして出たときも評判だったファンタジーゾーン。敵の宇宙人も含めてどれもみな可愛いキャラ，皆さんもヒロイックファンタジーを楽しみましたか？

マイコン歴1年半という作者の吉田さんは，友人との合作ですが，としてこのファンタジーゾーンから3曲送ってきてくれました。その中から，パーカッションやドラムの音がノリのいいHot Snowをご紹介しましょう。BASICからそのままrunして聞いてください。

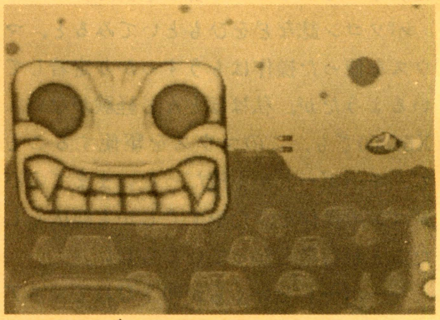
X-BASICです

3画面横並びの3倍角画面で話題になったダライアスと同じ筐体を使ったタイトーの横スクロール型アクションゲーム，それがニンジャウォーリアーズ。昨年の前半に出されたゲームですからお馴染みの人も多いでしょう。そのニンジャウォーリアーズのメインテーマがこのDuddy Mulkです。ボコーダーボイスや津軽三味線に代表されるサンプリング音をふんだんに使ったサウンドは，ゲーマーの間ではずいぶん人気を集めました。

作者の土井さんが送ってくれた5曲の中から，今回はこの曲を採用させていただきました。X-BASIC上でrunして聞いてください。それでは，1989年も皆さんの自信作をお待ちしています。



ソーサリアン


ファンタジーゾーン

リスト1　ソーサリアンオープニングテーマ

```
10 'Sorcerian Opening Theme    By Z.Nishikawa    (C)FALCOM
20 DEFINTA-Z:DEFSNG V
30 CLEAR&HFBB0:TEMPO0:CLS0:TEMPO0:"SOUND"
40 A1$="L8R2RDGAB2RGAB>C<BAB4.>F+4E4.<B4.>F+4 E4.<B&BBABG2{EF+G}2L2GF+GA&A1
50 B1$="L8R1G>DAB&B2CEBA&A2<EB>F+G&G2 <DB>F+G&G2<C+B>C+E&L2E<DDDO1P3I2D1&D2
60 C1$="L1RGE2.D+4G   G E  E2D2  E2F+2&F+
70 C2$="L1RBG2.F+4B   B G  G2F+2 G2A2&A
80 C3$="L1RDC2.<B4>F+ F+<B>C2<A2>C2D2&D
90 E$="L8R1D2.&DD DDDD4.D4 F+4.F+4.F+4 F+4.F+4.F+4"
100 E1$=E$+"V9 {>BGE GEC+ EC+<G >C+<GC+}1 V6O4L2DDD^1D1R2
110 E11$=E$+"V9 K1{>BGE GEC+ EC+<G >C+<GC+}1 V6O4K4L2DDD^1D1R2
120 E2$="L8R1G2.&GG GGGG4.G4 D+4.D+4.D+4 D+4.D+4.D+4 RV9K2{>BGE GEC+ EC+<G >C+<G
R}1 V6L2O4R4.GG^1G1R2
130 F1$="L8R1 G2&GGDG A4.B4.A4 G4.G4.EF+ G4.G4GF+E G1 D2 D2 D2D1 R2
140 '
150 A2$="O3L8R2A@4&B@20AG4>C4<BA&AG&GB&B2A@4&B@20AG4 >E-4.F&FC4<B&B2>DEEF F4<B>E
```

```
4D4E&E2<AB>CD D4F+A4C4<B&
160 B$="O1L8G>DAB&B2<G>E-G>C&C2<<G>DAB&B2"
170 BB$="G>DAB&B2<EB>EG+&G+2<A>EAB&B2<DA>DF+&F+2
180 B2$=B$+"P3<G>E-G>C&C2 O1P1"+BB$
190 B3$=B$+"L8V16O3K5P1A4GG&GE-4D32E32D16 P2V14O1K10"+BB$
200 C4$="L1O3GE-G E-GG+ EF+
210 C5$="L1O3BG B L8V16K10P2A4GG&GE-4D32E32D16 L1K5@V115P3BBAA
220 C6$="L1O4DC D C DE CD
230 E3$="^2D1E-2&E-8E-4.D1 E-2&E-8G4.D1E1E1 A4.F+2G8
240 E4$="^2G1G2&G8G4.G1    G2&G8B4.G1G+1C1  F+4.E2D8
250 F2$="O6L2 D.D4 E-4.F4F+4G8& G1 >C4.D+4C4<B8& B1>F2.&F8E8& E1 D+&D+8C4<B8&
260 '
270 A3$="L8B2B2 B2G+EG+B.B4.A4.E4 BAGA4B>C4< B4.A4.G4 BAGA4B>CD& D2.&D8D8& D2GF+
EF+ G1G1
280 B4$="L8<G>DGB<F>CFA<EB>DF+G+2<A>EGA&A2 <A>EGA&A2<F>CFA&A2 <F>CFA&A2 <DA>CD&D
A>CD&D1<<G>CDEG>CDEG2.R4
290 C7$="V13L2DCEE C4.C4&C4. C4.C8&C2 C1 C4.C2 G8& G2.&G8F+8&F+1V12G1G1
300 C8$="V13L2GFAG+E4.E4&E4. E4.E8&E2 F1 F4.F2 A8& A2.&A8A8& A1>V12C1<B1
310 C9$="V13L2<BABBB4.A4&A4. B4.A8&A2 A1 A4.A2>D8& D2.&D8D8& D1 V12D1D1
320 E$="L8<G>DGB<F>CFA<EB>DF+G+2<A>EGA&A2 <A>EGA&A2<F>CFA&A2 <F>CFA&A4.^2>D1D8&D
1
330 E5$=E$+"  ^0O3V8G>CDEG>CDE^1G2&G1
340 E6$=E$+"K1^0O3V8G>CDEG>CDE^1G2&G1
350 E7$="^1L2DCEEC4.C4&C4.C4.C8&C C1 C4.C G8&G.&G8F+8&F+1 K0^0V7O2L8G>CDEG>CDE^1
G2&G1
360 F3$="B1>E1<C4.C4.C4 C4.C2&C8 C1 C2.&C8 G1F+1&F+8>C1<B1
370 '
380 PLAY "T95I1O3V16P1"+A1$;:PLAY A2$;:PLAY A3$;
390 PLAY ":I1O3V16K10P2"+A1$;:PLAY A2$;:PLAY A3$;
400 PLAY ":I4O5V10P1S2,2,0,10=1H3"+F1$;:PLAY F2$;:PLAY F3$;
410 PLAY ":I1O1V14P1"+B1$;:PLAY "I1P1"+B2$;:PLAY B4$;
420 PLAY ":I1O1V14K10P2"+B1$;:PLAY "I1P2"+B3$;:PLAY B4$;
430 PLAY ":I1O3@V115"+C1$;:PLAY C4$;:PLAY C7$;
440 PLAY ":I1O3@V115"+C2$;:PLAY C5$;:PLAY C8$;
450 PLAY ":I1O4@V115"+C3$;:PLAY C6$;:PLAY C9$;
460 PLAY ":K0O4S0,0,0,0=1^2V5"+E1$;:PLAY E3$;:PLAY E5$;
470 PLAY ":K4O4S0,0,0,0=1^2V5"+E11$;:PLAY "K3"+E3$;:PLAY "K2"+E6$;
480 PLAY ":K0O4S0,0,0,0=1^2V5"+E2$;:PLAY E4$;:PLAY E7$
490 END
500 LABEL"SOUND"
510 MEM$(&HB190,36)=HEXCHR$("2C 00 72 72 32 32 15 10 14 10 1F 11 1F 1F 00 0E 00
0E 00 08 00 08 00 32 00 32 00 00 00 00 00 00 00 00 00 00")
520 MEM$(&HB1B4,36)=HEXCHR$("3B 00 32 04 66 04 1E 1E 28 09 5C 5C 5C 49 00 00 00
00 00 00 00 00 02 02 01 02 00 00 00 00 00 00 00 00 00 00")
530 MEM$(&HB1FC,36)=HEXCHR$("FB 53 72 02 31 01 37 39 37 00 DF 4C CF 8C 05 07 02
90 43 00 00 00 E5 F5 24 16 00 00 00 00 0C C4 90 0B 02 00")
540 RETURN
```

リスト2 Hot Snow

```
1000 '*******************************
1010 '*          F a n t a z y   Z o n e           *
1020 '*                                              *
1030 '*               S T A G E   5                  *
1040 '*                                              *
1050 '*          - - -   H o t   S n o w   - - -     *
1060 '*                                              *
1070 '*               b y   吉田   健智              *
1080 '*******************************
1090 '
1100 PLAY INIT
1110 DIM PERC%(4,9),DRUM%(4,9)
1120 FOR I=0 TO 4
1130     FOR J=0 TO 9
1140         READ PERC%(I,J)
1150     NEXT
1160 NEXT
1170 DATA 52,15,0,0,0,0,0,0,0,0
1180 DATA 30,19,30,8,13,23,0,7,0,0
1190 DATA 31,14,28,11,1,10,0,5,0,0
1200 DATA 31,19,13,12,4,22,3,14,-3,0
1210 DATA 31,15,10,9,1,0,3,10,3,0
1220 FOR I=0 TO 4
1230     FOR J=0 TO 9
1240         READ DRUM%(I,J)
1250     NEXT
1260 NEXT
1270 DATA 60,15,0,0,0,0,0,0,0,0
1280 DATA 31,0,0,8,0,14,0,15,0,0
1290 DATA 31,15,17,12,2,17,0,0,0,0
1300 DATA 31,24,0,8,11,19,0,0,0,0
1310 DATA 31,19,16,12,2,0,0,0,0,0
1320 TONE PERC%,DRUM%
1330 A1$="t127@v105o5l16brbr>f&fr<brbb&b>f&fffo6y140,4@v114"
1340 'B1B$="drrdrrdrdrrdrrdd":B1$="q4t127o4l16@v120y97,31y113,2
8y81,30"+B1B$+"q8o4l8y141,4y97,1y101,0y105,0y109,0"
1350 B1B$="crrcrrcrcrrcrrcc":B1$="q4t127o5l16@v115"+B1B$+"q8o4l
8"
1360 D1B$="m1500crm2500crm1500crm2500cr":D1$="y7,49t127s9l8"+D1
B$
1370 D2$="y7,49t127s9l8m1500crm2500crm1500c16r.t254l16v14cv12cv
9cv7c8rt127l8"
1380 A2$="@27d-rfra-d-rcre-rcfrcr"
1390 AA2$="d-r<b->d-r<b->rcre-rcffc"
1400 C2$="@27t127o5@v112l16y142,4a-ra-ra-a-ra-ra-ra-a-ra-r"
1410 CC2$="f+rf+f+rf+ra-ra-ra-a-a-a-"
1420 B2$="@11@v115d-&d-16<a-&a-16>d-c&c16e-&e-16<a-"
1430 BB2$="f+&f16c+&c+16f+g+&g+16>c&c16e-"
1440 E2B$="crrcrrcrcrrcrrcc":E2$="t127l16o2q1v15"+E2B$
1450 '
1460 '
1470 F4B$="dede":F4C$="abab":F4$="r8t254v13o6l32"+F4C$+F4C$+F4C
$+F4C$+">"+F4B$+F4B$+F4B$+"dedet127"
1480 A4$="d-rd-rfd-rd-rd-rd-f+rf+rfrd-"
1490 C4$="a-ra-ra-a-rb-rb-rb-b-rb-ra-ra-"
1500 D4$="m1500crm2500crm1500crq6l32v14cv9cv5cv3cv14cv9cv5cv3cv
14cv9cv5crv14cv9cv5cv3c"
1510 DD4$="r8s0m1500l32y6,5q1ccccl16rcl32ccccl16ccy6,0q8"
1520 E4B$="v13ev11d+v9dv6c":E4C$="v13c<v11bv8a+v6a>":E4$="crrcr
rcrcrrct254l32o4q8"+E4B$+"r8"+E4B$+"r8"+E4C$+"r8"+E4C$+"r8r4"
1530 EE4$="r4v13gv11f+v7fv5ev13gv11f+v7fv5er8"+E4B$+"v13gv11f+v
7fv5ev13gv11f+v7fv5e"+E4B$+"v13ev10d+t127"
1540 B4$="d-d-f16d-d-d-d-16f+f+fd-"
1550 A5$="rrf+rf+rrrfrrd-rrrr"
1560 AA5$="rrf+rf+f+rrfrre-rrd-"
1570 C5$="rrb-rb-rrra-rra-rrrr"
1580 CC5$="rrb-rb-b-rra-rra-rra-"
1590 B5B$="f+&f+16f+&f+16f+c+&c+16c+&c+16c+":B5$=B5B$
1600 E5B$="dedr":E5C$="abab":E5$="v12t127l8o7rrrrrrrt254l32"+E5
B$+"dede"
1610 EE5$="t127l8rrrrrrrt254l32"+E5B$+"de"
1620 A6$="rrf+rf+rrrfrrrd-rrr"
1630 AA6$="e-rrrrrrd-rrcrrrr"
1640 C6$="rrb-rb-rrra-rrra-rrr"
1650 CC6$="a-rrrrrra-rra-"
1660 B6$="a-&a-a-&a-16a-a-a-16b-a-"
1670 E6$="t127l8rrrrrrrl32t254"+E5B$+E5B$
1680 EE6$="t127l8rrrrrl6t254l32"+E5B$+"r16<"+F4C$+F4C$+F4C$+F4C
$+"abab"
1690 A7$="rrf+rf+rrrfrrrd-rrr"
1700 AA7$="e-rrrrrrrra-ra-a-rrr"
1710 C7$="rrb-rb-rrra-rrra-rrr"
1720 CC7$="a-rrrrrrrra-ra-a-"
1730 A8$="<b->rd-rr<b-rb->rd-r<b->d-d-<b->r"
1740 AA8$="<a->rd-rr<a-ra->rd-r<a-a-a->d-r"
1750 C8$="f+rf+rrf+rf+rf+rf+f+f+f+r"
1760 CC8$="frfrrfrfrfrffff"
1770 B8$="o3g-&g-16b-&b-16>d-e-e-d-<b-"
1780 BB8$="d-&d-16f&f16a-b-b-a-f"
1790 A9$="<b->rd-rr<b-rb->rd-r<b->ffe-r"
1800 AA9$="d-rrrrrrrrrra-a-rr"
1810 C9$="f+rf+rrf+rf+rf+rf+g+g+g+r"
1820 CC9$="frrrrrrrrra-a-"
1830 B9$="o3g-&g-16b-&b-16>d-e-e-d-<b-"
```

```
1840 BB9$=">d-&d-&d-&d-rr<a-16a-16>"
1850 D9$="y6,5ql116ccrcccrcccrcccl8y6,0q8"
1860 E9$="r16132o4q8t254"+E4B$+"qlo2v14c8c8c8o4q8"+E4C$+"qlo2v1
4c8c8c8o4q8"+"v13<bv11a+v7g+v5g>rrqlo2v14c8c8q8o4"+E4C$+E4C$+"t1
27"
1870 '
1880 '
1890 '
1900 '
1910 '
1920 '
1930 TONE LFO 6,0,1,180,1
1940 PLAY A1$,B1$,C1$,"y6,0"+D1$,E1$
1950 PLAY A1$,B1$,C1$,D1$,E1$
1960 PLAY A2$,B2$,C2$,D1$,E2$
1970 PLAY AA2$,BB2$,CC2$,D1$,E2$
1980 PLAY A2$,B2$,C2$,D1$,E2$
1990 PLAY AA2$,BB2$,CC2$,D1$,E2$
2000 PLAY A2$,B2$,C2$,D1$,E2$
2010 PLAY AA2$,BB2$,CC2$,D1$,E2$
2020 PLAY A4$,B4$,C4$,D4$,E4$
2030 PLAY ,,,DD4$,EE4$,F4$
2040 PLAY A5$,B5$,C5$,D1$,E5$
2050 PLAY AA5$,B5$,CC5$,D1$,EE5$
2060 PLAY A6$,B5$,C6$,D1$,E6$
2070 PLAY AA6$,B6$,CC6$,D2$,EE6$
2080 PLAY A5$,B5$,C5$,D1$,E5$
2090 PLAY AA5$,B5$,CC5$,D1$,EE5$
2100 PLAY A7$,B5$,C7$,D1$,E6$
2110 PLAY AA7$,B6$,CC7$,D2$,EE6$
2120 PLAY A8$,B8$,C8$,D1$,E2$
2130 PLAY AA8$,BB8$,CC8$,D1$,EE2$
2140 PLAY A8$,B8$,C8$,D1$,E2$
2150 PLAY AA8$,BB8$,CC8$,D1$,EE2$
2160 PLAY A8$,B8$,C8$,D1$,E2$
2170 PLAY AA8$,BB8$,CC8$,D1$,EE2$
2180 PLAY A9$,B9$,C9$,D1B$,E2B$
2190 PLAY AA9$,BB9$,CC9$,D9$,E9$
2200 GOTO 1960
```

リスト3 Duddy Mulk

```
  1 /*    ～～～～～～～～～～～～～～～～～  ざ にんじゃ うぉりあーず
  2 /*  T h e   N I N J A   W A R R I E R S  ざ にんじゃ うぉりあーず
  3 /*～～～～～～～～～～～～～～～～～～～  ざ にんじゃ うぉりあーず
  4 /*                                    ざ にんじゃ うぉりあーず
  5 /*" D a d d y   m u l k "-Main theme-R o u n d 1 & 6 -
  6 /*
  7 /*      by     T A I T O   Z U Z T A T A
  8 /*
  9 /*                            --Programed by Atsushi Doi
 10 M_INIT():FOR i=1 TO 8:M_ALLOC(i,10000):M_ASSIGN(i,i):NEXT
 11 DIM CHAR v(4,10)
 12 KEY 11,"m_tempo(200)@M":KEY 12,"m_tempo(160)@M
 13 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN    E . B A S S
 14 v={  27, 15,  2,  0,200,  0,  0,  0,  0,  3, 69,
 15 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS
 16       31, 21,  0,  8, 15, 21,  0,  6,  7,  0,  0,
 17       31, 15,  0,  8, 15, 35,  0,  9,  4,  0,  0,
 18       31,  0,  0,  6,  0, 26,  0,  0,  7,  0,  0,
 19       31,  8,  0, 10, 15,  0,  0,  1,  0,  0,  0}
 20 M_VSET(70,v)
 21 /*
 22 /*
 23 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN    S・D R U M
 24 v={  60, 15,  0,  0,  0,  0,  0,  0,  0,  3,189,
 25 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS
 26       31, 25,  5,  2,  0,  0,  0, 15,  0,  0,  0,
 27       31, 17, 14,  8,  5,  0,  0,  3,  0,  0,  1,
 28       31, 25,  0,  0, 15,  0,  0, 10,  0,  1,  0,
 29       31,  0, 15,  5, 15,  0,  0,  0,  0,  0,  1}
 30 M_VSET(71,v)
 31 /*
 32 /*
 33 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN    C Y M B A L
 34 v={  58, 15,  0,  0, 10,  0,  0,  0,  0,  3,188,
 35 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS
 36       31, 10,  5,  6,  0,  0,  0, 15,  0,  2,  1,
 37       31, 13, 18,  8,  8,  0,  0,  0,  0,  1,  0,
 38       31,  8,  9, 12,  6,  0,  0,  9,  2,  3,  0,
 39       31, 12,  9, 14,  8,  0,  0, 15,  1,  0,  0}
 40 M_VSET(72,v)
 41 /*
 42 /*
 43 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN    B・D R U M
 44 v={  60, 15,  0,  0,  0,  0,  0,  0,  0,  3,202,
 45 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS
 46       31, 10,  5,  6,  0,  0,  0, 15,  0,  0,  0,
 47       31, 20, 19,  8,  8,  0,  0,  0,  0,  0,  0,
 48       31, 16,  9,  7,  6,  7,  0,  0,  0,  0,  0,
 49       31, 12,  9,  9,  8,  0,  0,  0,  0,  0,  0}
 50 M_VSET(73,v)
 51 /*
 52 /*
 53 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN    T O M²
 54 v={  62, 15,  0,  0,  0,  0,  0,  0,  0,  3,188,
 55 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS
 56       31, 10, 10,  0, 15,  0,  0, 13,  0,  2,  0,
 57       31, 21, 17, 12,  7,  0,  0,  0,  0,  3,  0,
 58       31, 14, 11, 13,  8,  0,  0,  0,  7,  0,  0,
 59       31, 16, 12, 14,  9,  0,  0,  0,  3,  0,  0}
 60 M_VSET(74,v)
 61 /*
 62 /*
 63 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN    H and  C lap
 64 v={   0, 15,  3,  1,244,  0,  0,  0,  0,  3, 72,
 65 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS
 66       26, 10, 14,  5,  0,  0,  0,  0,  0,  0,  0,
 67       31, 10, 14, 10, 15,  0,  0,  0,  0,  0,  0,
 68       22,  5, 14,  6, 15,  0,  0,  1,  0,  0,  0,
 69       28, 15, 14, 10, 15,  0,  2,  8,  0,  0,  1}
 70 M_VSET(75,v)
 71 /*
 72 /*
 73 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN    S ynth. S ub
 74 v={  28, 15,  2,  1,210,  6,  0,  6,  0,  3, 83,
 75 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS
 76       31,  0,  0,  8, 10, 21,  1,  2,  7,  0,  0,
 77       31,  0,  0,  8,  0,  0,  1,  1,  3,  0,  1,
 78       31,  0,  0,  8,  0, 21,  1,  2,  3,  0,  0,
 79       31,  0,  0,  8,  0,  0,  1,  1,  7,  0,  1}
 80 M_VSET(76,v)
 81 /*
 82 /*
 83 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN    C H O R U S
 84 v={   5, 15,  2,  0,198,  7,  0,  5,  0,  3, 83,
 85 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS
 86       22,  4,  4,  0,  0, 32,  0,  4,  3,  3,  0,
 87       18,  0,  0,  8,  0,  0,  0,  4,  7,  0,  0,
 88       16,  0,  0, 12,  0,  0,  0,  3,  4,  0,  0,
 89       14,  0,  0, 14,  0,  0,  0,  2,  3,  0,  1}
 90 M_VSET(77,v)
 91 /*m_trk(1,"@77o3c1"):m_trk(2,"@77o3g1"):m_trk(3,"@77o3e1"):m
_play():end
 92 /*
 93 /*
 94 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
 95 v={   3, 15,  2,  0,202,  0,  0,  0,  0,  3, 77,
 96 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS  D a D i D a
 97       14, 10,  0,  5,  1, 58,  0, 14,  3,  3,  0,
 98       16, 10,  0,  5,  2, 26,  0,  0,  7,  0,  0,
 99       15, 10,  0,  5,  1, 37,  1,  7,  3,  2,  0,
100       15,  0,  0,  8,  0,  0,  0,  4,  7,  0,  0}
101 M_VSET(79,v)
102 /*
103 /*
104 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
105 v={  60, 15,  0,  0,  0,  0,  0,  0,  0,  3, 72,
106 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS  H - H close
107       31,  5, 28, 10,  6,  0,  0, 15,  0,  1,  0,
108       31, 18, 21, 13, 15,  0,  0, 15,  0,  1,  0,
109       29, 10,  0, 10, 15,  0,  0, 15,  0,  2,  0,
110       30, 20, 21, 15, 15,  0,  0,  1,  0,  0,  1}
111 M_VSET(80,v)
112 /*
113 /*
114 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
115 v={  60, 15,  0,  0,  0,  0,  0,  0,  0,  3, 72,
116 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS   H - H open
117       31,  5, 28, 10,  6,  0,  0, 15,  0,  1,  0,
118       31, 14, 21, 13, 15,  0,  0, 15,  0,  1,  0,
119       29, 10,  0, 10, 15,  0,  0, 15,  0,  2,  0,
120       30, 18, 21, 15, 15,  0,  0, 10,  4,  0,  1}
121 M_VSET(81,v)
122 /*
123 /*
124 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
125 v={  16, 15,  2,  1,200,  5,  0,  5,  0,  3, 75,
126 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS つがるじゃみせん
127       31, 10,  2,  5, 13, 27,  3,  2,  4,  0,  0,
128       31,  9,  5,  5, 10, 37,  2,  4,  1,  0,  0,
129       29,  8, 17,  4, 13, 16,  1,  1,  4,  0,  0,
130       29, 10, 10,  7, 10,  0,  0,  3,  3,  0,  1}
131 M_VSET(82,v)
132 /*
133 /*
134 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
135 v={  60, 15,  2,  0,201,  9,  0,  2,  0,  3, 83,
136 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS Synth.Lead
137       15,  4,  0, 14, 15, 32,  2,  1,  7,  0,  1,
138       18,  3,  0, 15, 15,  0,  1,  2,  3,  0,  1,
139       18,  3,  0, 14, 15,  0,  2,  1,  3,  0,  1,
140       18,  3,  0, 14, 15,  0,  2,  3,  7,  0,  1}
141 M_VSET(83,v)
142 /*
143 /*
144 /*
145 /*
146 /*
147 STR a[256],b[256],c[256],d[256],e[256],f[256],g[256],h[256]
148 STR da[256],db[256],dc[256],dd[256],de[256],df[256],dg[256],
dh[256]
149 KEY 14,"list 10000-59999@M"
150 M_TRK(1,"@83v11q718o5y48,00p3 t160")
151 M_TRK(2,"@83v05q818o5y49,40p3")
152 M_TRK(3,"@76v08q718o6y50,20p3")
153 M_TRK(4,"@79v12q618o3y51,40p3")
154 M_TRK(5,"@70v13q718o3y52,20p3")
155 M_TRK(6,"@71v15q818o1y53,00p3")
156 M_TRK(7,"@71v15q818o0y54,64p3")
157 M_TRK(8,"@73v15q818o0y55,00p3")
158 /*
159 /*  A  p a r t
160 /*
```

▶知能機械概論のチェルノブイリには驚いた。あるもんですねえ，こういうことって。
間島　謙一　(19)　東京都

```
161 /* SynLead / SynLead / SUB / DaDiDa / Bass / Snare / Snare /
 Bass Dra
162 h="ce-cf2&f"
163 b="ce-cf&":dh="f2"+h+"ce-cf4e-e-cc>b-b-<c2&c":a="ce-cf&"+dh
164  M_TRK(1,"r2"+b+"[do]"):M_TRK(2,"r2rce-c[do]fy49,40y48,0")
165 b=h+h+"ce-cf4gga-a-ggf2&f":dh=dh+b+a
166 b=h+h+"ce-cf4gga-a-b-b-<c":dh=dh+b
167  M_TRK(2,dh)
168 a=dh+"&v14c1&c1&c1r1>
169  M_TRK(1,a)
170 /*
171 g="ra-fd->a-<d-rg":f="ra-fc>a-<cra-
172 a="r1[do]@76v8q7o6|:"+g+"rge->b-g<e-r4"+f+"ra-fc<c>a-r4"+g+"
rgb-ge->b-<r4"+f+"|1ra-<c>a-fcr4:||2ra-ffa-fr
173  M_TRK(3,a)
174 /*
175 g="ra-ga-r2":g=g+g:f="ra-ga-b-a-ga-
176 a=g+"r1r1"+g+"r1r1"+g+f+"r1"+g+f
177  M_TRK(4,"r1[do]@79v12q6o3"+a)
178 /*
179 h="d-d-d-d-d-d-d- e-4e-e-e-e-e-e-":f="d-&"+h:g=h+"f4ffffff"
180 a=g+"f4fgga-":b=a+"a-g":a="d-&"+a:c=a+"a-b-":d=a+"gf
181 e=f+"f4gga-a-b-b-<c>
182  M_TRK(5,"r2.rd-&[do]"+b+c):M_TRK(5,d+e)
183 a="ffff":a=a+a+a
184 b=a+"fa-b-":c="f&"+a+"fa-b-<c>
185  M_TRK(5,b+c)
186 /*
187 a="@77q8o4
188 b="v11e-e-ddc4e-e-ddc4&cr4
189  M_TRK(2,a+b+"r"+b)
190 b="v12>|:ra-a-ggf4a-a-ggf4&fr4:|
191  M_TRK(4,a+b)
192 b="v10cc>b-b-a-4&a-r4<"
193  M_TRK(3,a+b+"rcc>b-b-a-4<"+b)
194 /*
195 da="r4c4"
196  M_TRK(6,da+da+"[do]@71o1|:15"+da+":|")
197 db="r4a4"
198  M_TRK(7,db+db+"[do]@71o0|:15"+db+":|")
199 M_TRK(6,"@72o0ra4a4.o1")
200 M_TRK(7,"@72o0p1rf4p2f4.o0")
201 M_TRK(6,"@71c4|:13"+da+":|")
202 M_TRK(7,"@71a4|:13"+db+":|")
203 dc="rbb4b4bb":dd="rbb4bbbb":a="|:3"+dc+":|
204 M_TRK(8,"b4b4bbrb[do]"+a)
205 M_TRK(8,dd+a+"rbb.b16bb4b")
206 M_TRK(8,a+dd+a)
207 M_TRK(8,"|:14b4:|bb16b16")
208 a="r8e&de&d e&d&c&>b< |:c&>b<:|
209 b="|:>g&f&e&d<:|":a=a+b
210 c="o4132
211 d=c+a+a+"r4"+b+a+a+"r418
212  M_TRK(6,"@74"+d):M_TRK(7,"@74"+d)
213  M_TRK(6,"@75o1a4"):M_TRK(7,"@75o1e4")
214  M_TRK(8,"@75r64a8.&a32.")
215 /*
216 /* B  p a r t
217 /*
218 a="c>b-<c>a-4b-4.a-4<
219  b="r4|:"+a+">e-<e-4c&c1&c1&c4"+a+"|1>ge-4c&c1&c1&c4:||2b-<c
>e-f4&f1ffgga-a-
220  M_TRK(1,"v11"+b+"b-"):M_TRK(2,"r@83o5v5"+b)
221 /*
222 a="r4{fga-b-<}4c2&v8c1&v7c1&v6c4r2.>v9":b="<e-c>a-fa-<ce-fe-
a-f<c>>r2
223 c="@76q7o5v9"
224 M_TRK(3,c+"y50,0r1r1 |:ra-gra-r4.grra-r2r1r1|1"+a+":||2"+b)
225 M_TRK(4,c+"y51,40r1r1|:rfe-rf r4.e-rrfr2r1r1|1"+a+":||2"+b)
226 /*
227 a="d-d-e-e-ffr e-4e-ffggr f4fgga-a-r":b=a+"f4fgga-a-<c>d-&
228  M_TRK(5,"|:3"+b+":|"):M_TRK(5,a+"f4ffgga-a-b-")
229 /*
230 b="r{cc}8{cccc}4
231 c="@74o4132r8 e&d&>c&b<|:4e&d:||:4c&>b<:||:4>a&g<:|18@71"
232 d="@71o1|:7"+da+":|"+b+"|:6"+da+":|"
233  M_TRK(6,d+c)
234  M_TRK(6,d+"r@v120{cc}8@v124{cccc}4v15{cccc cccc}2")
235 b="r{aa}8{aaaa}4"
236 d="@71o0|:7"+db+":|"+b+"|:6"+db+":|"
237  M_TRK(7,d+c)
238  M_TRK(7,d+"r@v120{aa}8@v124{aaaa}4v15{aaaa aaaa}2")
239 /*
240 dc="rbrbb4rb
241  M_TRK(8,"@73bbrbb4rb|:15"+dc+":|")
242 /*
243 /* C  p a r t
244 /*
245 a="b-32&<c8.&c32d-c4>b-4.a-4a-b-2
246 c="a-fr4.b-b-b-a-f
247  b="@83o4"+a+"&b-"+a
248  M_TRK(1,"|:v11"+b+"&b-"):M_TRK(2,"|:y49,40v5r"+b)
249 b="@79o3"+c+"r2."+c:d="@83o4f4ga-
250  M_TRK(1,"v11"+b+"rv12"+d+"b-:|"):M_TRK(2,"v12r64"+b+"r16.r6
4v5r"+d+":|")
251 /*
252 a="116|:6fr<d-d->:|fr<a-4.>
253 b="|:6e-rb-b-:|e-r<<d-4.>>
254 c="|:5fr<cc>:|fr<a-a-a-ra-a-r4>
255 FOR i=1 TO 2:M_TRK(3,a+b+c+c):NEXT
256 /*
257 a="d-4d-e-e-ff":b="e-4e-ffgg":c="f4fgga-a-":d=c+"r"
258  M_TRK(5,"|:d-d-e-e-ffr"+a+"a-"+b+"r"+b+"b-")
259  M_TRK(5,d+c+"<c>"+d+"f4fgga-a-<cc>:|")
260 /*
261 FOR i=1 TO 2
262  dd="|:6"+da+":|"
263   M_TRK(6,"@71o1"+dd+da+"r4cc"+dd+"r4c")
264  dd="|:6"+db+":|"
265   M_TRK(7,"@71o0"+dd+db+"r4aa"+dd+"r4a")
266  a="@74o4132|:4e&d:||:c&>b<:||:>a&g<:|18
267  FOR j=6 TO 7:M_TRK(j,a):NEXT
268  M_TRK(6,"@75o1a16a16"):M_TRK(7,"@75o1e16e16")
269  dc="rbb4b4":M_TRK(8,"|:3"+dc+"bb:|"+dc+"b4|:3"+dc+"rb:|")
270  M_TRK(8,dc+"rr64@75a16a32.@73")
271 NEXT
272 M_TRK(4,"@45v15o1|:r1r1r1r1 r1r1r1 r4c{cccc cccc}2r:|")
273 /*
274 a="@79o4e-d-cd-r4e-d-cd-re-d-cd-":a=a+"4"+a+"8.&d-32.
275 b="@83o5a-4.g4.f4."
276 b=b+"e-fc>b-<c4r"+b+"<c>b-<c>f<c4>
277  M_TRK(1,"|:v12q6"+a+"&d-64v11q7"+b+"r:|")
278  M_TRK(2,"|:v12q6r64"+a+"v6q8r"+b+":|")
279 /*
280 a="rf<c>f<e->frf<c>f<e->fr2":a=a+a
281 M_TRK(3,"18|:"+a+"r1r1r1r1:|")
282 /*
283 a="f<cc>b-b-a-a-b-
284 FOR i=1 TO 2
285  M_TRK(5,"|:16d-:||:16e-:||:8f:|"+a+"|:8f:|"+a)
286 NEXT
287 /*
288 a="@71o1"+da+da+da
289 b="@80o7rc@81c4":d="r2
290 c="{e&d&c&>b<}8":c="r4@74o4"+c+c
291  M_TRK(6,"|:"+a+b+a+b+a+c+a+"rccc16c16:|")
292 a="@71o0"+db+db+db
293  M_TRK(7,"|:"+a+d+a+d+a+c+a+"raaa16a16:|")
294  M_TRK(8,"|:16b4bbrbb4:|")
295  M_TRK(4,"|:r1r1r1r1r1r2.ccr1r2rccc16c16:|")
296 /*
297 /* D  p a r t  -  s y n t h . s o l o  -
298 /*
299 a="116|:8fa-b-<c>b-a-:|<cc8e-e-8ff8gg8a-rb-r"
300 M_TRK(1,"q7v12"+a):M_TRK(2,"q8v6r8"+a):M_TRK(3,"@83o5q6v3r4"
+a)
301 a="b-32&<c.>"
302 b=a+"b-a-f8":b=b+b+a+a:b="|:"+b+":|"
303  M_TRK(1,b):M_TRK(2,b):M_TRK(3,b)
304 b="b-a-fga-b-a-gfe-c>b-a-gfga-b-a-gfe-c>":c=a+b+"b-a-gfa-f<e
-
305  M_TRK(1,c):M_TRK(2,c):M_TRK(3,c)
306 a="ce-ce-fr":a="|:3"+a+a+"gra-r:|ff8gg8a-a-8b-b-8<cre-r>"
307  M_TRK(1,a):M_TRK(2,a):M_TRK(3,a)
308 a="|:20f<e->f<e-fr>:|f<e->f<e-"
309  M_TRK(1,a+"f4"):M_TRK(2,a+"f8"):M_TRK(3,a)
310 /*
311 a="|:4q7d-<q4d->:||:4q7e-<q4e->:||:4q7f<q4f>:|q7f<q4f>q7g<q4
g>q7a-<q4a->q7b-<q4b->q7
312 b="<d-d->b-b-a-a-gb-4b-a-a-gge-f4fe-e-ffrf4fe-e-fga-d-&
313  M_TRK(5,a+b+a)
314 c="d-d-e-e-ffa-e-4e-ffggb-f4fgga-a-<c>f4fgg<e-d-c>
315  M_TRK(5,c+"d-&"+c+"b-")
316 /*
317 a="@74o4132e&de&drr c&>b<c&>brr a&ga&grr e&de&drr18@71o0
318  M_TRK(6,"q8|:6"+da+":|"):M_TRK(7,"q8|:6"+db+":|")
319 FOR i=6 TO 7:M_TRK(i,a):NEXT:M_TRK(6,"<cc"):M_TRK(7,"aa")
320  M_TRK(6,"|:9"+da+":|"):M_TRK(7,"|:9"+db+":|")
321 b="@74o4r4{c&>b&a&g &f&e&d&c}4@71o1"
322 c="@74o4r4{c&>b&a&g<c&>b&a&g}4@71o1"
323  M_TRK(6,b+da+c+da+b):b=b+">":c=c+">"
324  M_TRK(7,b+db+c+db+b)
325 FOR i=6 TO 7:M_TRK(i,a+"@74o4{e&de&dc&>ba&g}4@71o0"):NEXT
326 a="rbb4bbrb":b="rbb4rb16b16bb"
327  M_TRK(8,a+b+a+a+"|:3"+a+b+":||:2"+a+":|")
328  M_TRK(4,"r1r1r1{ccrccrccrccrcrcr}1")
329  M_TRK(4,"|:4r1:|r2.c4r2.ccr2.c4{ccrccrccrccrcccc}1")
330 a="r4e4":M_TRK(4,"@75o1r64|:5"+a+":|r4e8.&e32.")
331  M_TRK(4,"@45o1r16c32c32|:14c16:|@75r64|:7"+a+":|r8.r32.@72p
1o0f4")
332 a="132@74o4rree |:8e&d:||:c&>b<:|>a&ga&g e&de&d@71o0
333  M_TRK(6,"<|:6"+da+":|"+a+"<|:7"+da+":|@72o0r4a4")
334  M_TRK(7," |:6"+db+":|"+a+" |:7"+db+":|r4r4")
335 a="|:3rb<@75e4>@73bb<@75e>@73b:|
336  M_TRK(8,a+"rbb4b4bb"+a+"rb<@75e4>@73bbb4")
337  sub()
338 /*
339 /*  F  p a r t
340 /*
341  M_TRK(1,"q7"+dh+"&c2.."):M_TRK(2,"r"+dh+"&c2.")
342 a="b-32&<c16.>":b=a+"b-16a-16f":c="|:2"+b+b+a+a+":|"+b+b+a+a
343  M_TRK(3,"@72o0v15f2r2r|:7r1:|@76v12o5"+c+"&<c4>b-a-ffa-b-")
:M_TRK(3,c+"&<c2..")
344 a="|:2ra-ga-r2:|r1":b=a+"r1
345  M_TRK(4,"@79v12q6o3"+b+b+b+a+"r2..")
346 /*
347 a="d-d-d-d-d-d-d-e-4e-e-e-e-e-e-f4ffffff4":b="d-&"+a
348 c="|:"+a+"b-b-a-a-gf"+b+"|1ffffa-b-d-&:||2|:6f:|
349  M_TRK(5,c)
350 /*
351 M_TRK(6,"@71o1|:15"+da+":|@72o0ra4a4.@71o1c4|:13"+da+":|@75o
1r4e4r4aa")
352 M_TRK(7,"@71o0|:15"+db+":|ra4.            |:14"+db+":|@75o
0r4a4r4ee")
353 a="rbb4b4bb":b="rbb4bbbb":d="|:3"+a+":|":c=d+b+d
354  M_TRK(8,c+"rbb.b16b@72o0p1f4p2f8@73o0")
355  M_TRK(8,c+"b4@75r64e4r16..@73b@75r64ee16..")
356 /*
357 M_TRK(1,"y48,64v15q8o5@19f8")
358 M_TRK(2,"y49,0v15q8o2@20f4")
359 M_TRK(3,"y50,40v15q8o4@34f4")
360 M_TRK(4,"y51,20v15q8o5@18f4")
361 M_TRK(5,"y52,40v15q8o2@62f8")
362 /*
363 /*  G  p a r t  -  s y a m i s e n  s o l o  -
364 /*
365 a="e-32&f."
366  M_TRK(1,"q7o5116@82v13y48,0|:8"+a+":|")
```

▶この前，念願かなって秋葉原に行ってきた。電車に揺られること4時間あまり。やっと秋葉原のホームに立った。ここまではよかったのだが，どの階段を上ろうが下りようがどこまでもホームだった。そのとき自分の方向音痴を知るとともに，東京の広さを知った。

有田 直之 (17) 福島県

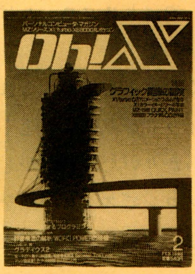
```
367  M_TRK(2,"q7o5l16@82v13y49,40r8|:6"+a+":|")
368  M_TRK(3,"q7o5l16@82v6y49,100r8r|:6"+a+":|")
369 d="{e-fe-}8c":a=d+"8":b=a+"c8c8"
370 c=b+b+a+a+a+d+"e-"
371 a="f8e-e-e-8ccc8>b-b-b-8<e-e-e-8ccc8>b-b-b-8a-a-a-8f8"
372 b="|:3{a-b-a-}8f8:|{a-b-a-}8e-e- e-8fff8a-a-a-8<cce-8>f<e-
373 e="|:8e-32&f-.:||:a-8a-8a-8f8:||:3b-8a-8:|b-8<e-e-e-8ccc8>b-
b-b-8a-a-f8b-b-
374 f="b-8a-a-a-8fff8e-e-c8>e-e-f8cce-8ffa-8e-e-f8a-a- b-8ffa-8b
-b-<c8>b-b-<c8e-8
375 FOR i=1 TO 3:M_TRK(i,c+a+b):M_TRK(i,e+f):NEXT
376 /*
377 a="{e-fe-}8c8":b="{>b-<c>b-}8
378 c="|:"+a+"c8c8"+b+"f8f8f8<:||:4"+a+b+"a-8:|<|:8f8:||:e-8f8f8
f8:|
379 d="|:4{b-a-fb-a-f}4f8<c8>:|
380 e="<c>b-a-f"
381 f="|:4b-a-f"+e+e+e+"<c>:|
382 FOR i=1 TO 2:M_TRK(i,c+d+f):NEXT
383 f="|:3b-a-f"+e+e+e+"<c>:|b-a-f"+e+e+e
384 M_TRK(3,c+d+f)
385 /*
386 M_TRK(8,"v15@73o0|:20b4:|")
387 M_TRK(7,"v15@74o2|:11e4:|e8")
388 a="@74o4{e&d c&>b<}8
389  M_TRK(6,"r1r1r2.."+a+"@71o1|:17"+da+":|")
390  M_TRK(7,a+"@71o0|:17"+db+":|")
391 M_TRK(4,"v15r1r1r2.@45o1c16c16@72o1f2r2")
392 M_TRK(5,"v15r1r1r1  @72o1a4r2.")
393 a="b4b4b4bb":b="b4b4b4b4":c="b4r4b4b4"
394  M_TRK(8,"|:7"+a+":||:"+b+":||:"+a+":||:4"+c+":||:40b:|{bbbb
bb}1")
395 M_TRK(4,"@72|:6r1:|f2r2r2.@45{ccc}8c8r1r2.{cccc}4|:R2.{CCC}8
C8:|")
396 M_TRK(5,"@72|:6r1:|a4r2.")
397 a="@74o4r4{e&d c&>b a&g}8{e&d&c&>b}8>@71
398 M_TRK(6,a+" |:3"+da+":|r116|:6c:|18|:r4c4"+a+":|")
399 M_TRK(7,a+">|:3"+db+":|r116|:6a:|18|:r4a4"+a+">:|")
400 a="e&d e&d c&>b&a&g a&g&f&e<":b="e&d e&d c&>b a&g<"
401 c="@74o4132|:4"+a+a+b+":|
402 FOR i=6 TO 7:M_TRK(i,c):NEXT
403 M_TRK(4,"r1r1@72o1f2r2r1r4r16f2f2f2r8.")
404 M_TRK(5,"|:7r1:|  a4r2.r1r4r16a4r4a4r4a4r4..")
405 a="{e&d c&>b<c&>b g&f}4{e&d&c&>b<<}8
406 b="@74o4|:"+a+"r8:|"+a+"r16{ee}16|:5{e&d}16:||:{c&>b<}16:|>{
g&f}16
407 M_TRK(6,b):M_TRK(7,b)
408 /*
409 M_TRK(6,"l16|:@71o1cccc cccc o7@80c@81cc@80c@81cc@80c@81c:|l
8")
410 M_TRK(7,"l16@71o0|:aaaa aaaa r2:|l8")
411 a="{c&>b< e&d&c&>b&a&g}4{g&f&e g&f&e g&f&e}4<
412 b="@74o4|:3"+a+":|{e&d e&d e&d e&d}4{c&>b< c&>b< c&>b&a&g}4
413 M_TRK(6,b):M_TRK(7,b)
414 /*
415 M_TRK(4,"@74o2|:{cccc cccc}2r2:|r8@72o1f2f2f2r4.")
416 M_TRK(5,"r1r1r8@72o1a4r4a4r4a4r4r4.")
417 /*
418 FOR i=1 TO 8:M_TRK(i,"l8"):NEXT
419 sub():M_TRK(3,"r8")
420 FOR i=1 TO 8:M_TRK(i,"[loop]"):NEXT
421 M_PLAY():END
422 /*
423 /* E  &  H  p a r t
424 /*
425 FUNC sub()
426 M_TRK(1,"l8@83o4y48,0v10"):M_TRK(2,"l8@83o4y49,40v10")
427 b="rfff ffff":a="q8|:3"+b+"ffgga-a-b-b-:|"+b
428  M_TRK(1,a+"fgga-a-b-b-b-<"+a+"ffggq7ce-cf&")
429 b="r|:9d-:|e-e-ff":c="rcccc cccc"
430 a=b+"ff"+b+"gg"+c+"ce-e-ffff"+c
431  M_TRK(2,a+"e-e-fffff<"+a+"ce-e-v5rce-c")
432 /*
433 a="|:16d-:||:16e-:||:31f:|
434  M_TRK(5,"@70v13q7l8o3y52,20"+a+"f"+a+"d-&")
435 /*
436 a="r2@79o3v12r4f4e-fa-<c>":a=a+"r2"+a:b="r2.rc
437  M_TRK(4,"l8@72o0p2v15f2r2|:6r1:|")
438  M_TRK(4,"@45o1r{ccc}8c{ccc}8c{ccc}8@72o0p1f4p2f2"+a)
439  M_TRK(4,"r1r1r1v15@45o1|:4{ccc}8:|")
440  M_TRK(4,"@75o0r64a8.&a32.r@72f")
441 /*
442 a="|:7r2.g4:|
443 M_TRK(3,"q8@45o2v15l8"+a+"r1"+a+"r2@75o1e4r8")
444 /*
445 a=da+da
446  M_TRK(6,"l8@72o0a4@71<c4r4c4|:6"+a+":|")
447 b="{e&d c>&b a&g<}8":c="{>e&d&c&>b<<}8
448 d="@74o4r8|:2"+b+c+":|"+b
449  M_TRK(6,d+"@72o0a4a4@71<c4r4c4|:6"+a+":|")
450 e="@74o4|:{e&d c&>b a&g a&g<c&>b<e&d}4:|
451  M_TRK(6,e+"@71o1c4r@72a")
452 /*
453 a=db+"r4aa"
454  M_TRK(7,"l8@71o0|:7"+a+":|")
455  M_TRK(7,d+"r4@71o0|:7"+a+":|")
456  M_TRK(7,e+"@71o0a4r@72a")
457 /*
458 dc="|:7rbb4bbv13b4v15:|
459  M_TRK(8,"@73l8"+dc+"|:4b4:|"+dc+"b4b4@75o1a@73o0bbb")
460 ENDFUNC
```

バックナンバー案内

BACK ISSUES

ここには1988年2月号から1989年1月号までをご紹介しました。現在，1987年4，8，11，1988年1，2，3，4，6，7，8，9，11，12，1989年1月号までの在庫がございます。バックナンバーおよび定期講読のお申し込み方法については，本文172ページを参照してください。

1988



2月号
特集　グラフィック画像の冒険
X1/turboCGアニメ/トリフォニーで立体モデル
X68000グラフィックデータ/QUICK MZ PAINT他
X68000あなたの知らない世界　辞書構造/WORD POWER
マシン語体操1・2・3　Lispインタプリタ(1)
● NEW Z-BASIC詳報　その名はZ-BASIC
● LIVE in '88　グラディウス2
● SHORT ACCESS　THRILLING/POMカードポーカー
全機種共通システム　シューティングゲームELFES



3月号
特集　コンピュータサウンド"楽"入門
X1/turbo MIDIインタフェイスの製作
MZ-2500 Super Keyboard/VIPサウンドデータ公開
Oh!X LIVE SPECIAL 組曲「Ys」/Raspberry Dream他
THE SOFTOUCH　Might and Magic/Hyper UD
オブジェクト指向のゲームプログラミング
X68000BASIC入門　奇襲アニメ作戦
X68000あなたの知らない世界　未公開IOCSの解析
全機種共通システム　構造型コンパイラ言語SLANG



4月号
特集　不思議の国のゲーム学
決定！　1987年度GAME OF THE YEAR
ピコピコゲーム春場所/GAME REVIEW 10本他
新製品　X68000ACE-HD/カラースキャナCZ-8NS I
X68000あなたの知らない世界　microEMACSの移植
● MZ-700　SPACE BLUSTER FX
● LIVE in '88　Moonlight Serenade/Long Night 他
全機種共通システム　デバッギングツールTRADE
シミュレーションウォーゲームWALRUS



5月号
特集　BASIC入門「再検証」
BASICの歴史と意義/栄光のHuBASIC
黄金のBASIC入門プログラム/プログラミング用語集
ミュージックプログラマへの道/レイトレーシング
特別企画　言わせてくれなくちゃだワ
● 新製品　X68000ACE/ACE-HD
● LIVE in '88 GET WILD/BOOM BOOM/SDI
● SHORT ACCESS 3Dボクシング/マシン語データ文生成
全機種共通システム　シューティングゲームELFES



6月号　創刊6周年記念
特集　システム環境を考える
8ビットパソコンの開発環境/Human68kのシステム環境/システムを読むためのアセンブラ入門
特別企画　究極の8ビットパソコン 8RON計画
THE SOFTOUCH　X68000用日本語ワープロEW 他
● 付録「あぶない福袋」
マシン語体操1・2・3　番外編　Lisp80入門
X68000BASIC入門　捨て身のミュージック
全機種共通システム　構造化言語SLANG入門 他



7月号
特集　実践C言語からの誘惑
入門C言語/実録Cプログラミング/XBAS to C
THE SOFTOUCH　ソーサリアン/ゼリアード/アルギースの翼/SUPER大戦略/3大麻雀ソフト 他
● Oh!X LIVE in '88/SHORT ACCESS
新連載　C調言語講座PRO-68K　まずはprintfより始めよ
あなたの知らない世界　OS-9/X68000／Sampling PRO-68K
全機種共通システム　構造化言語SLANG 入門(2)
マルチウィンドウドライバMW-I



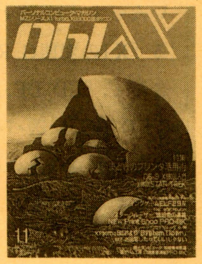
8月号
特集1　真夏の夜の数値演算
コンピュータの数値表現/応用グラフィック歪められた光/AD PCM音の数学/数値演算プロセッサ用ドライバ 他
特集2　MIDIサウンドプログラミング
MIDIの基礎とボードの製作/MIDI対応シーケンサ
THE SOFTOUCH　新連載　われら電脳遊戯民　他
猫とコンピュータ第26回　ボクはかぐや姫？
新連載　Z80マシン語ゲーム工房
全機種共通システムマルチウィンドウエディタWINER



9月号
特集　半期に一度のグラフィックバザール
CGアニメの手法入門/ワイヤフレームによる3D/X68000スプライト/画像処理の基礎知識/turbo RAY TRACER/MZ-2500用グラフィックエディタDMACS
THE SOFTOUCH　C-TRACE68/SAMPLING PRO-68K他
C調言語講座PRO-68K(3)　謎の低次元グラフィック
MIDI活用テクニック(2)　割り込みによるMIDI通信
Z80マシン語ゲーム工房(2)　応用への基礎固め
全機種共通システム　ラインエディタTED-750/WINERの拡張



10月号
特集　百花繚乱ゲームバトルロイヤル
最新ゲーム総登場　ハイドライド3/A列車で行こうII/たんば/熱血高校ドッジボール部/フルスロットル他
MZ-700用SPACE HARRIER
● Oh!X LIVE 1974(16光年の訪問者)/瑠璃色の地球/二人のゼネレーション/バッハのアリア
MIDI活用テクニック(3)複数の音源を操るテクニック
C調言語講座PRO-68K(4)/Z80マシン語ゲーム工房(3)
全機種共通システム　SLANG用拡張ライブラリ/MANKAI



11月号
特集　いまどきのプリンタ活用術
メカニズムを理解しよう/制御コード/文字と図形の混在印字/拡大文字のスムージング/外字登録ツール/S-H COPY/グラフィックのモノクロ出力/X68000のCOPYキー/オリジナル印刷キット/試用レポート
THE SOFTOUCH　NEW Print Shop PRO-68K 他
OS-9/X68000入門(1)　OS-9ってなに？
● STAR TREK for X68000
全機種共通システム　シューティングゲームELFES IV



12月号
特集　パソコンはいま音楽の領域へ
なぜ自動作曲か/心地よい雑音の話/和音の読み方/美しい響きの要素/4分音符は歌い始める/古くて新しい音楽形式/FM音源の仕組み/Melody Box/MusicBASIC
● さよなら Live in '88　バッハ イタリア組曲ほか6本
● Oh!X　1周年記念特別企画「ちょっとあぶない福袋」
OS-9/X68000入門(2)　OS-9のオペレーション環境
Z80マシン語ゲーム工房/C調言語講座PRO-68K
全機種共通システム　ソースジェネレータSOURCERY

1989



1月号
特集　いきなり初春からハードウェア
デジタル回路入門/電子サイコロ/乱数発生器/X1turboバンクメモリ拡張/X68000用CP/M-80システム他
1988年度GAME OF THE YEAR　ノミネート作品発表
● MZ-2500用　Hyper Game Book
● LIVE in'89　エンデューロレーサー/アルルの女
● ようこそ，セガ・メガドライブ！！
C調言語講座PRO-68K/Z80マシン語ゲーム工房
全機種共通システム　パズルゲーム LAST ONE/FLICK

QUESTION and ANSWER

Q **X1の市販のソフトのなかにはPCGを使っていないのに画面をスクロールしているものがありますが，X1には隠れたスクロール機能のようなものがあるのですか？**

神奈川県　三村　和子

A 今月は「味っ子PRO-68K」こと私，西川善司がお答えします。

おおっ，数少ない女性読者からの質問ですね。まず，質問の答えを先にいってしまいましょうか。X1には「隠れたスクロール機能」なるものはありません。がっかりした人も多いかな。でも，ときどき不思議なくらいグラフィックを速くスクロールさせているゲームがありますね。この「タネあかし」の前に，X1のグラフィックRAM（以下G-RAM）について考えてみましょう。

PC-88SRなどには「ALU」という面白い機能が付いています。これは簡単にいうとマシン語1命令でRGB複数のプレーンのデータを処理できるというものです。88などはG-RAMをバンク切り換えによってメインメモリにもってくるので，G-RAMをLDIR，LDDRといった命令でブロック転送が可能なのです。X1はG-RAMがI/Oにくっついていますね。メインメモリとI/O。すでに，この時点で，88にX1は（グラフィックの速さという点では）負けています。

それはG-RAMにひとつデータを送る場合を見れば一目瞭然です。88では「LD (HL),A」で命令の処理速度は7クロック。X1では，「OUT (C),A」としなくてはならないので，12クロックで88に5クロックの差をつけられています。もしRGB3プレーンをスクロールさせる場合はどうなるでしょう。88はALUにより，RR (HL)で15クロック。X1においてはIN A,(C)：RRA：OUT (C),Aの3プレーン分となり，28クロック×3＋α（アドレス計算にかかる時間）で，88の5倍以上遅いわけです（同時アクセスモードなるものがX1にはありますが，あれを行っている間は，割り込みは禁止しなければいけないし，データは読み出せないし，「ALUは比較データなるものを読み出せる」。なによりも，このモードにするまでの手続きが面倒臭いので，あまり使いものにはなりません）。

このほかにも原因はあります。X1のG-RAMのマップをハード解析書（祝一平著『試験に出るX1』など）で見てもらえばわかるように，実に他機種と比べて奇妙な並びをしています。アドレス計算が少々面倒臭いんですよね。これが，業界で恐れられている「X1の変態G-RAM」なのです。88の場合だと，「LD DE,????：LD HL,????+80：LD BC,転送バイト数LDIR」で，縦1ドットスクロールになってしまうのですから。

しかし，88とて，パソコン。X68000のようにハードウェアスクロールが付いているわけではないので，やはりスクロールはそれほど得意ではないのです。

さて，前置きが長くなりましたが，これでパソコンはグラフィックがそれほど速くないということがおわかりいただけたかと思います。では，パソコンでスクロールをするのによい方法はないのでしょうか。実はあるのです。

いわゆる「部分スクロール」というやつがそのひとつです。この方法を用いているゲームは，たぶん「ソーサリアン88」，「アルギースの翼」，「ロマンシア」，「ザナドゥ」，「リバイバー」，「ガンマ5」などではないかと思います。では，この「部分スクロール」とは，いったいどんなものなのかを説明しましょう。

ソーサリアンのような横スクロールのケースを検討することにします。まず，画面はチップ（画面を構成する最小単位）で構成します。ここでは，図1のようなもの(21×8）を考えます。図1の「囲み1」が現在表示している画面です。画面はこれから，いままさに右にスクロールしようとしているところだとします。「囲み1」の右に縦1列ありますが，これがスクロールした場合に右に新しく登場するべきデータです。

図1ではわかりやすくするためチップをアルファベットで表していますが，実際には「A」などはレンガだったり，「D」は木の葉だったり，「F」や「E」は地面の絵だったりするわけです。この部分は，画面をいかにチップで構成しているかをわからないように作成するかという，デザインのセンスによって大きく変わってくるところです。「リバイバー」や「ソーサリアン」はなかなか上手な部類かな。さて，このチップですが，「ソーサリアン」や「リバイバー」などのゲームに使われているチップと同じ，「16×8ドット（2バイト×8バイト），8色（RGB3プレーン）」とします。

さあ，ここで画面が右にスクロールしました。「囲み2」がスクロール後の画面です。もしここで，すべてのチップを描き換えたとしたら，2×8×3（プレーン）×21×

図1

①
```
                                        I   I
          B B B         D             I I   I
G       B B B B B     D D D         I I I   I
G G   B B B B B B B   D D D       I I I I   I
G G G   A A A A A     D C D     I I I I I   I
H H H   A A A A A       C     I I I I I I   I
F F F F F F F F F F F F F F F F F F F F F   F
E E E E E E E E E E E E E E E E E E E E E   E
```

②
```
                                      I I
        B B B         D             I I I
      B B B B B     D D D         I I I I
G   B B B B B B B   D D D       I I I I I
G G   A A A A A     D C D     I I I I I I
H H   A A A A A       C     I I I I I I I
F F F F F F F F F F F F F F F F F F F F F
E E E E E E E E E E E E E E E E E E E E E
```

8＝8064バイトのデータを転送しなくてはなりません。

しかし，そんなことが必要でしょうか。「囲み1」と「囲み2」をよーく見比べてみてください。たとえば，地面にあたる「F」や「E」はまったく変わりませんし，家（のつもり）の壁「A」や屋根の「B」は左右の端が変わっただけで中心のほうは「A」「B」のままです。右の山（のつもり）の「I」はどうでしょう。山の輪郭が左に動いただけで，真ん中のほうは「I」のままです。どうです，わかってきましたか？　そう，そのとおり。描き換える必要のあるチップだけ描いてやるのです。余計なところはいじらずに。

では，いまの例だとどうなるんでしょう。描き換えの必要があるチップの個数は……33，33個ですか。それでは，全部描き直す場合と比べてみると2×8×3（プレーン）×33＝1584ですから，全部描き直しの時間の20％弱ですむわけですよ。こ，これは，うーまーいーぞー!!　と，これが「部分スクロール」です。しかし，X1にはPCGがありますから，あまりこの方法はX1では用いられていないようです。

ソフトハウスなどでは，これをさらに発展させた「オーバー付き部分スクロール」とか，「3段階部分スクロール」などが開発されています。オーバーというのは，ほら，「イース」みたいに木や柱の影に主人公や敵が隠れたりする処理のことです。3段階部分スクロールというのはその名のとおり画面が3重スクロールしてしまうやつです。まあ，これらはチップを管理するワークに多くの意味を持たせてやることで比較的容易に実現できますからここでは触れません。

これで，みんなはスクロール「通」!!（ブラボーおじさんの声で）　（西川　善司）

Q　X68000のROMデバッガについての使い方，必要なソフトおよびハードウェアについて教えてほしい。一部の雑誌には，ほかにRS-232Cを付けたコンピュータが1台必要と書いてあったのですが，どうなのでしょうか。ちなみに私はMZ-2531を持っているので，それを使いたいと思っています。

青森県　坂井　一弘

A　ROMデバッガを使うには，RS-232Cを備えたコンピュータ1台と，RS-232Cのクロスケーブル，そして，ターミナルマシン上で動作する「RS-232Cを介して相互に文字列を送ったり受け取ったりするプログラム」が必要です。

ターミナルにMZ-2500を使うのであれば，新たにプログラムを用意しなくても，BASICのTERM文を使えばよいでしょうし，通信ソフトがあればそれを流用してもかまいません。

ROMデバッガは，標準の状態では使用できる状態にありませんので，SWITCHコマンドでROMデバッガを起動するように設定しておきます。具体的にはSWITCHのプロンプトに続いて，

　－DB　＝　ON

と入力します。この設定を解除するには，その逆に，

　－DB　＝　OFF

とします。さらに，接続する2台のマシン間での通信パラメータを揃えます。このとき注意しなければならないのは，X68000側の設定はSPEEDコマンドではなく，SWITCHを使うということです。

以上の設定がすんだらターミナルマシン側のプログラムを待機状態にして，X68000を再起動します。するとターミナルのディスプレイにROMデバッガのタイトルメッセージが表示されるはずです。以後，インタラプトスイッチが押されるか，バスエラーなどのエラーが発生すると，ROMデバッガが起動し，制御がターミナル側に移ります。

ROMデバッガでは，DB.X（XCや福袋ver2.0に含まれるデバッガ）のコマンドのうちファイル入出力を除くほとんどのコマンドが使用可能ですから，それらを駆使してバグの要因を探してください。なお，Hコマンドで簡単なコマンド一覧が出ますから，プリントアウトして使うのがよいでしょう。

ROMデバッガからX68000へ制御を戻す方法はいくつかありますが，簡単なのはリセットをかけることです。特にインタラプトスイッチを押して起動し，かつ，デバッグ中にレジスタの値を変化させていなければ，

　－G

でX68000側へ戻ることができます。エラーが原因で起動した場合などはメモリ上に，DOSコール＿EXITを実行するプログラムを書き込んで実行するという手もあります。

DB.Xは単体でもターミナルと併用しても使え，いつでも制御をOSに戻すことができ，ソース上でのシンボルを使ってデバッグを行えるなどの点で，ROMデバッガよりも使いやすいと思われますが，デバイスドライバなどのデバッグにはROMデバッガが欠かせません。その意味でも，また，X68000にあらかじめ用意されている機能を生かすという意味でも，ROMデバッガはこれまで以上に，もっと活用されてもよいように思います。

最後になりましたが，Oh!X 1987年12月号の「X68000あなたの知らない世界」に，ターミナルを使わずにROMデバッガを利用する方法を含む簡単な紹介記事がありますので，可能であればそちらも参考にしてみてください。　（村田　敏幸）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒102 東京都千代田区
九段南2-3-26井関ビル
㈱日本ソフトバンク出版部
「Oh！X質問箱」係

愛読者プレゼント

プレゼントの応募方法

●

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1989年２月18日の到着分までとします。当選者の発表は1989年４月号で行います。

1

アンス・コンサルタンツ　☎092(522)6347



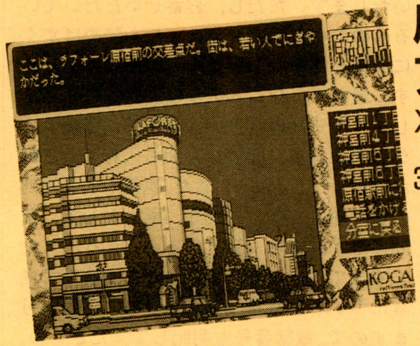
彩CRONE

X68000用　5″2HD版
1名

3Dレイトレーシングツール彩CRONEを1名の読者に。独自のモデリング方式で入力も簡単，リアリスティックな画像が楽しめる。春休みの目玉はこれで決まり？

2

ブラザー工業　☎052(824)2493



原宿アフターダーク

X1turbo用　5″2D版
(2ドライブ専用)
3名

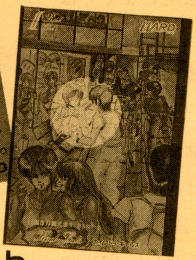
原宿を舞台に繰り広げられるミステリーアドベンチャー。華やかな町の裏側に広がる謎また謎を，果たしてあなたは犯人逮捕に向けて収束させることができるか？

3

ハード　☎03(837)1893



a.口説き方教えます



b.カインドゥ・ギャルズ



c.ダブル・ヴィジョン

X68000版　5″2HD版　各3名

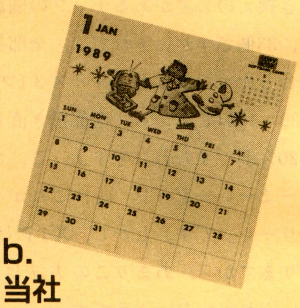
ハード社から，かの「口説き方教えます」はじめ３種類のソフトをそれぞれ３名に。３つともX68000対応。どれにもかわいい女の子がたくさん出てきて楽しいですよ。

4

ライフボート　☎03(293)4711



a.ライフボートオリジナルカレンダー



b.当社オリジナルカレンダー

各10名

ライフボートと当社の1989年度オリジナル卓上カレンダーを各10名の読者にプレゼント。机の上やディスプレイの上に置いて役立ててください。

12月号プレゼント当選者

特別モニタープレゼント[1] CZ-601C（東京都）中里宗弘[2]CZ-8PC3(群馬県）武田勝幸[3]CARD PRO-68K（神奈川県）鯛富之（愛知県）蕨野耕一郎（福岡県）上村健一　**愛読者プレゼント**　[4]沙羅曼蛇（埼玉県）武笠淳一（千葉県）西方孝一（静岡県）遠藤慎弥（富山県）三矢明良（岡山県）安藤雅司[5]ドラゴンスピリット（東京都）荻野浩司（宮城県）佐藤康治（山梨県）木下理[6]サンダーフォースII（千葉県）鶴岡英昭（広島県）狭間学（和歌山県）佐田佳子[7]ラスト・ハルマゲドン(神奈川県）雷有二(静岡県）寒河井友明（岡山県）森本幸宏[8]アークス（埼玉県）川村隆行（北海道）朝生隆義（兵庫県）服部元宏[9]缶詰（愛知県）山口寛憲[10]謎の福袋 a.(三重県）池田国治（鹿児島県）岩城圭一 b.(山梨県）相田幸宏（山口県）藤井義裕　他３名 c.(栃木県）林章（福岡県）平田省吾　他３名 d.(大阪府）平本雅之（兵庫県）石井仁　他８名 e.(静岡県）森下保（兵庫県）浅田英政　他８名　（敬称略）

以上の方々が当選されました。おめでとうございます。品物は順次発送いたしますが，入荷状況などにより遅れることがあります。また，公正取引委員会の告示により，このプレゼントに当選された方は，この号の他の懸賞には当選できない場合がありますのでご了承ください。

郵便はがき

料金受取人払

麹町局
承認
3018

差出有効期間
1990年 9 月30
日まで

1 0 2

(受取人)

東京都千代田区
九段南2-3-26井関ビル

(株)日本ソフトバンク

Oh!X 編集室行

□□□-□□

電話

住所

氏名

年齢

職業・勤務先
学校・学部・学年

●編集室へのメッセージ

今月号の特集について

いちばん良かった記事	興味のなかった記事
これから載せてほしい記事内容	本誌以外にお読みのパソコン雑誌

推薦する市販ソフト
ソフト名：
推薦理由：

あなたの愛機は(所有機種に○印をつけてください)　ない
X1(マニアタイプ,C,D,F,G,twin)　X1turbo(model 10,20,30,40,II,III,Z,ZII,ZIII)
MZ-(80K/C, 1200, 700, 1500, 80B, 2000, 2200, 2500, 2861)
X68000(600, 601, 611)　その他
FD(　基)　TAPE　QD　HD(　MB)　プリンタ(　)

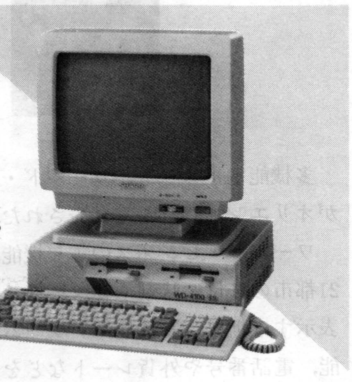
年齢　歳　パソコン歴　年　男・女　プレゼントNo.

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

ニュー書院の新製品
WD-4100シリーズ
シャープ


WD-4100S

ワープロのニュー書院シリーズの新製品WD-4100S/Dがシャープより発売された。価格はそれぞれ335,000円と380,000円。プリンタは別売り。

WD-4100シリーズは，14インチディスプレイと3.5インチフロッピーディスク2基を採用。作成した文書ファイルは，他の3.5インチFDDを持つ書院シリーズと共用でき，また標準装備のMS-DOSテキストファイルコンバータで変換し，パソコンで使用することも可能。

基本辞書約12万語，AI辞書約5万例を持ち，最大10文節(最長60文字)の連文節変換が可能。画面のガイドに従いながらタイピング練習のできる機能もついている。

さらに，表計算ソフト「書院カルク」，カード型データベース「書院カード」を装備，オプションで書院カルク用のアプリケーションも用意されている。そのほかの別売品は，ゴシックフォントファイルやハンディスキャナ，通信ソフト，RS-232Cインタフェイスなど。

なお，WD-4100Dには通電転写プリンタも接続可能。

〈問い合わせ先〉
シャープ㈱　☎06(621)1221，03(260)1161

電子手帳用ハンディプリンタ
CE-60P
シャープ

電子手帳PA-8500/7000/6500/6000用のハンディプリンタCE-60Pがシャープより発売中。価格は26,000円。

印字部が本体からは分離しており，いろいろな紙に印字が可能。文字の大きさは，縦・横とも3，6，9，12mmの4通り，計16種類の明朝体。また，カラーリボンは6種類用意されており，黒・赤・青・茶(各600円)のほか金・銀(各700円)の印字もできる。

オプションの毛筆体カートリッジCE-61Mは16,000円。

〈問い合わせ先〉
シャープ㈱　☎06(621)1221，03(260)1161


CE-60P
(電子手帳に接続中)

プリンタ新機種
SP-500，FP-850/1050
セイコーエプソン

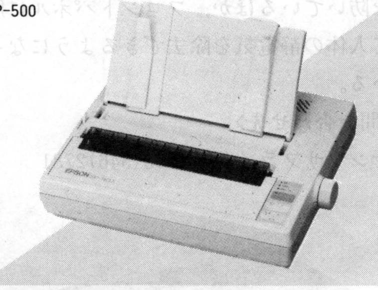
セイコーエプソンは，データやリストの打ち出しに適した9ピンシリアルインパクトドットプリンタSP-500，FP-850/1050の3機種を発売した。

SP-500は印字桁数80，速度はドラフトで150cps，NLQで25cps。入力データバッファ約3Kバイト。価格は60,000円。

FP-850は印字桁数80，速度はドラフトで220cps，NLQで45cps。入力データバッファ約8Kバイト。価格は110,000円。

FP-1050は印字桁数136，速度はドラフトで220cps，NLQで45cps。入力データバッファは約8Kバイト。価格は140,000円。

〈問い合わせ先〉
セイコーエプソン㈱　☎0266(52)3131

SP-500


パーソナルモデム
PM-2400F
富士通

2400/1200bps対応・全二重パーソナルモデムPM-2400Fが富士通より発売された。価格は59,800円。

ヘイズATコマンドに対応し，また通信プロトコルMNPクラス5の採用によるデータ圧縮を行い最大実効速度4800bpsを実現したという。

MNPのエラーフリー機能により，文字化けなどの再送処理を自動的に実行。CCITTのV.22などMNP以外に対応しているモデムとももちろん通信できる。

〈問い合わせ先〉
富士通㈱　☎03(216)3211


PM-2400F

パソコン用オールインワン電源
マスターピースSSI-8501
サンワサプライ

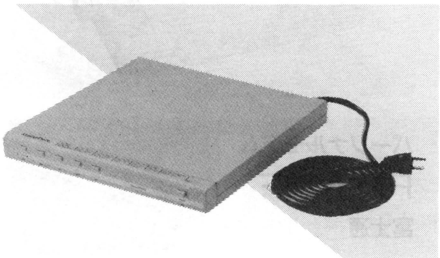
パソコン周辺の電源を一括してコントロールできるオールインワン電源，マスターピースSSI-8501がサンワサプライから発売

された。価格は17,800円。

コンピュータ，モニタ，プリンタ，AUX1，AUX2の５つのコンセントを持ち，マスタースイッチですべての電源をオンオフできるほか，各機器個別にもコントロールできる。

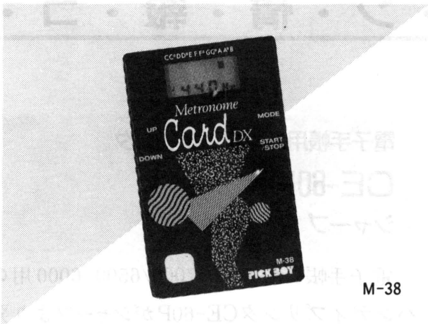
また，サーキットブレーカーやIEEE規格のサージ電子カット回路を内蔵して，高電圧・ラインノイズなどによる機械の誤動作を防いでいるほか，フロントパネルに触れて人体の静電気を除去できるようになっている。

〈問い合わせ先〉
サンワサプライ㈱　☎03(546)2781



マスターピースSSI-8501

メトロノームカード
M-25/M-38
ナカノ



M-38

メトロノームカードM-25とM-38がナカノから発売された。縦86×横54×厚さ５mm，重さ40ｇ，正確なテンポ・ビートをクオーツ精度で刻む。

M-25はテンポ範囲30から240，０から６のリズムが選べる。価格は2,500円。

M-38はテンポ範囲40から208，リズムは0,2から6に加え三連符にも対応。12のクロマティック基準音は８段階にピッチシフトでき，より広範囲の楽器のチューニングを可能にした。価格は3,800円。

リチウム電池を使い，連続して約240時間使用できる。

〈問い合わせ先〉
㈱ナカノ　☎03(863)1558

多機能電子カード
ワールド・ビジネス
オリエント時計



ワールド・ビジネス

多機能電子カード，ワールド・ビジネスがオリエント時計から発売された。

ワールド・ビジネスの主な機能は，世界21都市の日付・時刻やサマータイムなどを表示する世界時計機能，アラームセット機能，電話番号や外貨レートなどを登録しサ

Again Watch

BTRON，いよいよ発進？

超架空パソコンといわれるBTRONがいよいよ現実のものとなる日が近づいた？

TRON計画の推進団体であるトロン協会はさきごろ「TRONプロジェクト国際シンポジウム」を開催したが，これによりある程度BTRONパソコンの実在ぶりが明らかになった。

まず，ペーパーマシンである仕様書が年末年始にまとまり，出版される。また一部業界筋によると，ごく限定した形ながら，松下電器などのメーカーから希望者に対してBTRONマシンが有償提供され出したという。今春からは，この対象提供がさらに広がるらしい。聞くところによると，肝心のBTRON-OSの完成度はかなりのレベルまで高まっているらしい。すでに，例の教育用標準パソコン用に，ソフトハウスなどでアプリケーションソフトの開発作業が始まっている。もっとも，こちらはVAXでクロス開発しているとのことだが。

これまで長々と「そろそろ出る」といわれ続けてきたBTRONマシン。私も情報がアップデートされるたびに，このコーナーで紹介してきたが，以上の話はいよいよか，と思わせる材料ではある。一部でいわれているように宗教性が強い製品だけに，はっと気がつくと信者だけがこっそりと隠れTRONユーザーに……なんてことはないだろうが，やや危ない展開も含めて，やはり今年のキーワードとしてBTRONは浮上するのであった。

富士通の32ビット家庭用パソコン

これも現実の話ながら，やや架空のパソコンと化してきた。そもそもこの32ビット家庭用パソコンというのは，富士通が昨年末に東京ドームで開く予定だった「電脳遊園地」で衝撃のデビューをさせるべく準備してきた秘密兵器。ところがくだんの事情により，電脳遊園地の開催を無期延期したため，自動的にこのパソコンのデビューも遅れてしまったのが事の真相である。したがって自粛期間中は謎が続く。

この秘密兵器，Oh！X誌上でもいろいろな観測がなされているようだが，基本線としては，巷間でいわれているように，X68000の対抗機種とみて間違いない。なんでもCPUにi80386を使っているのがミソだとかで，機械の性格としては，これまでの８ビット機FM77AVシリーズを拡大発展した製品だそうだ。もっとも，386を使うからには77AVとのソフトの互換性はないだろうし，メガドライブのようにテーブルゲームのソフトをそのまま移植するわけにもいかないだろう。

それにしても富士通は電脳遊園地で南野陽子などタレントをたくさん呼んで，話題だけで日本電気を圧倒，はっと気がつけばこの32ビットパソコンで市場を席捲する計画だったという。

こういうささいな事ひとつとっても，日本電気の妖邪パワーが市場にたち込めているのだろう。

ちなみに富士通は自粛期間が終われば，電脳遊園地計画を再開する予定でいるという。

とはいえ，巷間では「３月に云々」という噂も聞く。架空のパソコン，などと書いたが，このニューマシンが早く顕現することを期待したい。

ーチでアルファベット順に呼び出せるメモ機能，スケジュール管理機能，電卓機能などで，外貨早見表つき。

価格はメモリの容量などによって4,800円から6,800円まで。名刺サイズのカード型で，重さは25ｇ。

〈問い合わせ先〉

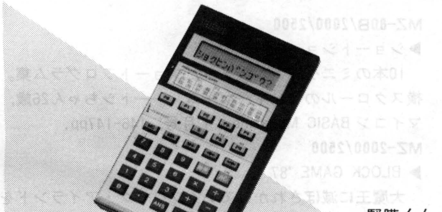
オリエント時計㈱　☎03(255)1580

摂取栄養量専用計算機

腎臓くん

カシオ計算機

カシオ計算機は，摂取栄養量が簡単な操作で計算できる専用機，腎臓くん(29,800円)を発売した。



腎臓くん

この計算機は，310種類の食品データ，腎臓疾患や高血圧症の食餌療法に役立つ専用計算などを内蔵。食べた食品とその量を入力すれば，摂取栄養量や必要な栄養の種類・量が指示されるので，特に栄養に関する知識がなくとも食生活の管理ができることを特徴としている。独立メモリもついて通常の電卓としても使用可能。

サイズは縦148.5×横83.2×厚さ11.5㎜，重さ116.5ｇ。

〈問い合わせ先〉

カシオ計算機　☎03(347)4830

INFORMATION

テレホンアドベンチャー

アミューズメントクラブ・プロダクツ

アミューズメントクラブ・プロダクツのテレホンアドベンチャー「地層階級王国2」が2月7日まで実施される。番号は，

東京03(236)9988，札幌011(821)9000

新潟025(267)7000，長野0262(35)8000

京都075(751)7700，倉敷0864(34)5550

広島082(252)0000

〈問い合わせ先〉

㈲アミューズメントクラブ・プロダクツ

☎0422(44)4321

BOOK

MZプログラム大全集

電波新聞社

マイコンBASIC Magazine DELUXEとして『MZプログラム大全集』が発売された。MZシリーズ用のプログラムが満載。B5判，358ページ，1,500円

〈問い合わせ先〉

電波新聞社　☎03(445)6111



MZプログラム大全集

AXは高級パソコンだったのか?

先日，「AXコンベンション」という展示会があり，AXパソコンが一堂に展示された。

会場で私はズラリと並ぶAXパソコンに圧倒されてきたのだが，ここでふと，AXパソコンがいつの間にか高級ビジネスワークステーションへの道をひた走っていることを再認識してしまった。

AXパソコンがIBM-PC/ATの日本語版互換機であることはおそらく誰でも知っているはずだ。すでにオリジナルメーカーのIBMが廃盤にした過去のパソコンであり，決していまさら新製品としてありがたがるようなシロモノではない。ところが，さすがにAXを推進しているメーカーが三洋，三菱，シャープ奈良グループ，沖電気工業，台湾エイサーなどのマイナーリーグだけあって，そのあたりを見事に錯覚しているらしい。32ビットCPUを使い，ハードディスクを搭載し，GSPを使って高速高解像度モードで武装するなど，ものものしいことこの上なし。

もちろん，その路線が絶対に誤っているかといえば，必ずしもそうではあるまい。しかしながらAXパソコンの当初の趣旨が，PC-9801共同対抗機種であったことを考えると，やはりオフコンまがいの機種に走り，CADワークステーションを指向することは路線を逸脱していると言っても間違いないところだろう。AT互換機なのだから，すでに海外で安く量販している機種をちょっと改造すればそれで製品化できる。現にたいていのメーカーはそうしている。エプソンの98互換機よりも安く作ることは至って簡単なはずなのだ。

ハードディスクなんて別売品を買えば20Mバイトでも7，8万円なのだから，そんなものを内蔵してやみくもに高くするよりは，フロッピードライブだけでもいいから，本体価格をもっと安くしてほしい。さすがに私もMSXより安くしろとはいわないが，せめてエプソンの98互換機よりは安くしないと，AX計画自体がつぶれてしまう。

TVゲーム近況

最後にテレビゲーム機のお話。長い間，噂にのぼっていた任天堂のスーパーファミコンが今年後半戦から登場することになった。今年下半期の話題を独占することはほぼ間違いない。

鬼のいぬ間に，というわけではなかろうが，日本電気のPCエンジンとセガのメガドライブが快調に売れている。とくにPCエンジンはこの年末年始で50万台を出荷する計画だそうで，累計でもすでに百万台を突破したとか。またこわごわ発売したCD-ROMドライブも約6万円という高額ながら，当初予定した4万台をすべて売りつくすなど，こちらもなかなかのものだ。一方のセガも，この年末年始で20万台くらいは売るつもりらしい。なんといっても，68000搭載が売り物で，テーブルゲームを手軽に移植できるのは強みだ。

ファミコン，PCエンジン，メガドライブのうちどれを買うかという時代ではなく，全部買う人もけっこういるようだ。まあ300万円以上の自動車がポンポンと景気よく売れる時代なのだから，ゲーム機くらいは売れて当然か。

金余り現象とか。私の周りにはそんな金持ちは見あたらないのだが，とにかく裕福でけっこうなことだ。　(K.T.)

FILES Oh!X

このインデックスは，タイトル，注記——筆者名，誌名，月号，ページで構成されています。いよいよ寒さも本格的です。ところで，読者の皆さんの中にはコタツでパソコンしている人っているんですか？

参考文献

I/O　工学社
ASAHI パソコン　朝日新聞社
ASCII　アスキー
テクノポリス　徳間書店
ファミコン通信　アスキー
POPCOM　小学館
マイコン　電波新聞社
マイコン BASIC Magazine　電波新聞社
LOGIN　アスキー

一般

▶ハイテク地獄耳
　シャープのミニ書院 WD-580/350F/290F/70/75ほか数メーカーのワープロを紹介，また X1turboZ IIIなど最新機種の情報を収録。——編集部，POPCOM，1月号，148-153pp.

▶ CD-I 最新レポート 4
　いよいよ発売間近の CD-I を，実物を見せながら紹介。具体的な性能データを付けて解説している。——編集部，POPCOM，1月号，228-230pp.

▶なんでも Q&A　シャープ MZ シリーズ編
　シャープのラップトップパソコン AX286L シリーズの価格，特徴，電子システム手帳とのデータ互換性について。——シャープ，マイコン，1月号，424-425pp.

▶コンピュータ・ミュージック講座 I　MIDI でなにができる？
　近頃の音楽制作に欠かせない MIDI について。6回集中講座の第1回。——山本数生，マイコン BASIC Magazine，1月号，55-57pp.

▶ FM 音源おもしろセミナー　オペレータってなあに？
　FM 音源を扱ううえでの基礎知識，オペレータ，モジュレータなどについて説明している。——川野俊充，マイコン BASIC Magazine，1月号，61-63pp.

▶奇跡の机上印刷術に挑戦
　個人でも比較的簡単に手に入れられるようになった DTP ソフト。まずは遊びから始めてみよう。——編集部，LOGIN，12月2日号，172-179pp.

▶図解　世界のコンピュータちゃん
　SUNワークステーションを例にマルチウィンドウを解説。巨大コンピュータ用語の独断的解説が面白い。——編集部，LOGIN，12月16日号，256-257pp.

MZ-80K/C/1200/700/1500

MZ-80K/C/1200/700/1500

▶誌上公開質問状
　MZ-1500で使えるデータレコーダの紹介など，MZ シリーズ関係の質問に答えている。——編集部，マイコン BASIC Magazine，1月号，70p.

▶ HUNGRY MOUSE
　あなたはハラペコねずみ。2匹の猫に捕まらないようにチーズをとって，自分の巣穴に運びこめ。面を追うごとに猫や障害が強くなる。——TECHNO DORAEMON，マイコン BASIC Magazine，1月号，138-139pp.

MZ-700/1500

▶ NUMBER MOLE
　ちょっと変わったモグラたたき。作物を喰われないようにモグラ（数字）を推理する。——まっぴー，マイコン BASIC Magazine，1月号，140-141pp.

▶ BRAVE KNIGHT'S ADVENTURE II
　ルーレット式に戦う簡単なRPG。ある目的を達成するために君はスライムと戦わなくてはならない。——松平義弘，マイコン BASIC Magazine，1月号，142-143pp.

MZ-1500

▶おつかれ清掃員
　あなたは未来の清掃員コロリン。街に落ちているゴミ袋を拾って面クリア。宇宙人にぶつかるとアウト。——山野辺太郎，マイコン BASIC Magazine，1月号，144-145pp.

MZ-80B/2000/2500/2800

MZ-80B/2000/2500

▶ショートショート
　10本のミニゲームが集まったショートプログラム集。横スクロールのワンキーゲーム。——トシちゃん26歳，マイコン BASIC Magazine，1月号，146-147pp.

MZ-2000/2500

▶ BLOCK GAME '87
　大魔王に滅ぼされかけているブロック・アイランドを救うために，あなたは戦う。敵は3種類，全20面の大作。——石田学，マイコン BASIC Magazine，1月号，148-150pp.

MZ-2500

▶ BUG FIRE
　BUG FIREのMZ-2500への移植版。オールマシン語のアクションゲーム。数々のアイテムを使って逃げ遅れた坑夫を脱出させる。なお，このゲームには MZ 用のパーソナルCP/Mが必要。——伝説阪神，I/O，1月号，168-177pp.

▶ PUSHER Part2
　テンキーでEDY君を動かしてブロックを運ぶパズルゲーム。フラッピーみたい。——山田秀男，マイコン BASIC Magazine，1月号，151-153pp.

▶ FANTASY ZONE　親玉出現のテーマ
　SEGAのファンタジーゾーンのBGMプログラム。——吉田健智，マイコン BASIC Magazine，1月号，212-213pp.

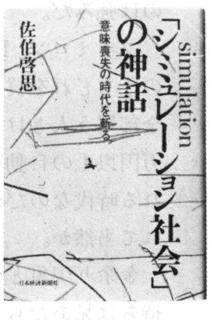
X1/X1turbo/Z

X1シリーズ

▶ MIAH
　オールマシン語，グラディウスタイプの横スクロールゲーム。パワーアップで各面のボスキャラを倒せ！——A．岩ちゃき，マイコン，1月号，264-273pp.

▶誌上公開質問状

新刊書案内



「シミュレーション社会」といわれてもわかりにくい。これを展開すると"現実感が希薄で，すべてが何かのシミュレーションのように感じられる社会"という感じでいいだろう。高度情報化社会なるものと同義としてもとりあえずはよい。本書はそのシミュレーション社会を東西の学者さんらの書物から引用などしながらわかりやすく解き明かす本である。が，それだけではない。著者は，シミュレーション社会を象徴する，"情報化が新しい価値観を生み出す"という発想やら，"シミュレートされたものとしての人間"という概念を解き明かしておいて，それを否定する反論を常に付加する。その反論がもっと圧倒的な論理をもって迫っていたら衝撃的だったろう。
　要するに，確かに現実感が希薄な社会だし，コンピュータやネットワークの発達が「軽さ」と「流れ」を助長し，「電子化されたニヒリズム」が幅を利かす時代ではあるけれども，それは「『表層』を情報と情報化のイメージによって制御しよう」としているだけであって，巻き込まれてはいけないよ，となるのである。（K）

「シミュレーション社会」の神話
佐伯啓思著　日本経済新聞社刊
B6判　242ページ　1,300円　☎03(270)0251

X1Gmodel 30に付属のディスクBASICでCZ-8FB01V1.0を起動する方法や,X1turboZで描いた絵をカラー印刷可能なプリンタの紹介など,X1関係の質問に答えている。——編集部,マイコンBASIC Magazine,1月号,71p.

▶ CARDS
ただの神経衰弱かと思ったら,なんとカードが勝手に動いてしまう。ちょっとひねったアクション神経衰弱ゲーム。——川口浩,マイコンBASIC Magazine,1月号,189-190pp.

▶プッシュ!!
プッシュ君を動かして氷を押し,3つの宝を取る。パーフェクトでクリアしないと12面以降に進めない。——UGN SOFT,マイコンBASIC Magazine,1月号,191-192pp.

▶ F-1 SPIRIT
ゲームミュージックプログラム。——長尾啓一,マイコンBASIC Magazine,1月号,214-217pp.

▶ NEW SOFT
ピラミッドソーサリアン,Might & Magic Book II,殺人は手紙にのって,などを紹介。——編集部,LOGIN,12月16日号,18-21pp.,34p.

▶ SOFTWARE REVIEW
ソーサリアン追加シナリオ2の戦国ソーサリアンを紹介する。——ルークとハン・ソロ,LOGIN,12月16日号,46-47pp.

X1turboシリーズ

▶競馬ゲーム
オールBASICの競馬ゲーム。——井上靖隆,I/O,1月号,187-188pp.

▶ハードウェア・シャープX1turboZ III
新製品のひとつとして,シャープのX1turboZ IIIの主なスペックや特徴などについて紹介している。——編集部,ASAHIパソコン,1月号,16p.

▶なんでもQ&A X1/X1turbo/X68000シリーズ編
X1turboZの2HDドライブを2DDモードで使うためのディップスイッチの変更方法について。——シャープ,マイコン,1月号,427p.

▶ SNAKE WALL
1度通ったところを通ったり,壁にぶつかったりしてはいけない。2人で遊べる,いわゆるスネークゲームである。——末包博文,マイコンBASIC Magazine,1月号,193-194pp.

X68000

▶自己平方フラクタル
自作の固定小数点演算ルーチン(整数部16ビット,小数部16ビット)を使った自己平方フラクタルの描画プログラム。——杉本正勝,I/O,1月号,206-208pp.

▶グラフィックライブラリ
DMAを使ったXC用のグラフィックライブラリ集。LINE, COPY, SYMBOLなどのルーチンがサポートされている。——来夢雷人,I/O,1月号,218-227pp.

▶ SOFT BOX
シャープ発売のX68000用MIDI対応ソフトMusicstudio PRO-68Kの主な特徴,機能についての紹介。——編集部,I/O,1月号,258-259pp.

▶ X68K INFORMATION SHOP
現在開発中のX68000用トランスピュータボードの紹介。——編集部,ASCII,1月号,337-338pp.

▶ X68K ReportShop
ビーピーエスのエディタFINALのX68000版について。高速スクロール,画面分割機能,マクロ機能などの紹介。——宮本親一郎,ASCII,1月号,339-341pp.

▶ X68K Technical Shop
OS-9/X68000のAV環境をサポートするライブラリ群PSS(プレゼンテーション・サポート・システム)の中からCD-I互換の環境を実現する部分についての概要を紹介。——中山進,ASCII,1月号,342-344pp.

▶ GAMING WORLD
開発途中のデス・ブリンガーや,今夜も朝までPOWERFULまあじゃん2の紹介。——編集部,テクノポリス,1月号,12-13,24-25pp.

▶パソコンゲーム通信
現在発売中のソフトの画面写真,これから発売するソフトなどの紹介。——編集部,ファミコン通信,12月9日号,156-157pp.

▶ X68000ワールド
カサブランカに愛を,WARNING,今夜も朝までPOWERFULまあじゃん2,Musicstudio PRO-68Kと,X68000でアニメを作っている「プロジェクトチームDOGA」を紹介。——編集部,POPCOM,1月号,106-109pp.

▶ X68000マシン語入門
sls・rtr命令などのフラグ関係とI/Oや割り込みなどの特殊な操作に関する命令の解説,そしてIOCSコールによるグラフィックの描画を行っている。——高橋雄一,マイコン,1月号,184-196pp.

▶ LISP言語REVIEW(2)
LISP言語特集の1項目として,AI-68Kの特徴や付属するユーティリティソフトなどを紹介。——編集部,マイコン,1月号,334-335pp.

▶ FM音源110番
X68000FM音源のLFO機能や各パラメータについて説明している。——Yu-You,マイコンBASIC Magazine,1月号,64-65pp.

▶ SUPER BOUND BALL
ボールを操作して箱に当て,アイテムを集めるゲーム。コンストラクションつき。——株式会社マイケル商事,マイコンBASIC Magazine,1月号,195-197pp.

▶カルテット
SEGAのカルテットの1,2面BGMプログラム。——川野俊充,マイコンBASIC Magazine,1月号,204-206pp.

▶チャレンジ! X68000
発売が待たれる,ボスコニアン,ファンタジーゾーン,そしてパワーリーグをレポート。——川野俊充,マイコンBASIC Magazine,1月号,292-293pp.

▶ X68000新聞
めぞん一刻,三国志,アークス,パックマニア,太平洋の嵐,SPRITE PRO-68K,Musicstudio PRO-68K,彩CRONEなどの紹介とC-TRACE講座の第2回。——編集部,LOGIN,12月2日号,192-197pp.

▶最新ゲーム徹底解剖!!
今夜も朝までPOWERFULまあじゃん2を徹底解剖。またサンダーフォースIIやデス・ブリンガーなどを紹介。——編集部,LOGIN,12月16日号,204-207,220-225pp.

▶ X68000新聞
C-TRACE講座に加え,シャープが開発に着手したアナログジョイスティック,新作ゲームやMusicstudio PRO-68Kのデータ集の紹介など。——編集部,LOGIN,12月16日号,258-263pp.

▶ X68000新聞
話題の各新作ゲームの紹介,またC-TRACE講座はオブジェクトのデータの色の決定の仕方について。——編集部,LOGIN,1月6・20日号,254-259pp.

ポケコン

PC-1260

▶ DANGEROUS ZONE
PC-1245の古典的名作であるDANGEROUS ZONEのPC-1260への移植版。仮想画面を使ったシューティングゲームでオールマシン語,約5Kバイト。——小野博義,I/O,1月号,196-197pp.

PC-1245/1417G/1450

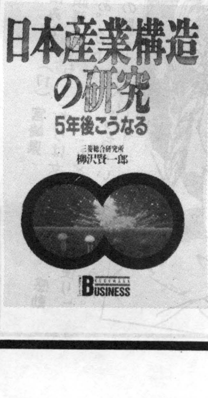
▶誌上公開質問状
PC-1417Gのマニュアルについて,またPC-1245とPC-1450のシステムサブルーチンの違いは,など。——編集部,マイコンBASIC Magazine,1月号,70-71pp.

PC-1501

▶ AGOCKの野望
くじで当たって勇者にされてしまった主人公。スライムやボスを倒しながら進む大作アクションRPG。——ほえっく,マイコンBASIC Magazine,1月号,201-202pp.

PC-E500

▶ポケコンスケジュール手帳
3カ月の予定管理ができるスケジュール手帳のプログラム。オールBASICでフローチャート付き。——折原美昭,マイコン,1月号,389-398pp.



日本産業構造の研究
著者は三菱総合研究所に勤務する経済学者。「情報化社会への移行」が叫ばれて久しい現在,それ(情報化)が社会のインフラストラクチャにどう影響していくかを,5年後の日本経済に焦点を当てて考察しているのが本書である。(やはり,いずれは重厚長大産業と軽薄短小産業の関係は逆転するのだろうか。)AI,光技術,超電導,核融合,バイオなど,話題の先端技術がやはり話題にされている。硬いタイトルだが内容は読みやすい。
柳沢賢一郎著 講談社刊
B6判 224ページ 1,200円 ☎03(945)1111



テレコンピューティング情報源'89年版
この本では,これからパソコン通信を始めようという人向けに,パソコン通信やオンラインデータベースについてその基本構造や概念を説明するとともに,現在アクセスできる主要なネットワークを紹介している。ビールス騒ぎでネットワークは戦々恐々としているようにマスコミでは報じていたが,電話回線による通信は,いまはまだそういうものだと考えれば,今後も参加する人間は増えていくだろう。
杉山勝行編 JICC出版局刊
A5判 272ページ 1,450円 ☎03(234)4621

STUDIO X ON AIR



FROM READERS TO THE EDITOR

さて，いよいよ冬も本番。スキー，スケート，温泉旅行(？)にと絶好のシーズンの到来です。風邪などひかないようがんばって遊びましょう。受験生の方はもう少しの辛抱。これまでの成果を存分に発揮して，みんなで楽しい春を迎えましょう。

◆今回発表された「MusicBASIC」を，さっそく打ち込んで使ってみましたがとても凄いですね。ソーサリアンを入力して聴き比べてみると，格段の違いです。でも，自分ではあまりプログラムをしないので，まだ十二分には使いこなせていません。今度はサンプル曲に，歌謡曲もあればいいとも思います。それと少し遅れてしまいましたが，Oh!Xの1歳の誕生日，おめでとうございます。これからも泉重千代さんに負けないくらいがんばってください。

富田　昌義（18）大阪府

富田君が早くこのMusicBASICを使って，音楽プログラムを投稿してくれるようになればいいなぁと，こちらも期待して待っていますのでガンバッテみてください。

◆「どんとこい！　ピコピコゲーム」を読んでいると，PC-6001を初めて買い，必死になってBASICを覚え，ピコピコゲームを作っては楽しんでいたことを思い出しました。いまではそのPC-6001は私の父の麻雀マシンとなっています。当時を思い出し，PCからX1turboにコンバートして今度投稿しようかと密かに思っています。

藤本　智弘（27）東京都

◆僕はまだパソコン歴が浅く，ゲームの作り方もイマイチ理解できません。でも12月号のピコピコゲーム「UFO来襲」は，かなり短かったので完成させることができました。ヤッター！

太刀川　仁（13）東京都

それはオメデトウ。太刀川君もこのときの感激を忘れずに，次回のピコピコゲーム開催のときには，ぜひとも応募してくださいね。パソコンを始めたばかりの方のほうが新鮮なアイデアで勝負できるので，もしかすると有利かもしれないしね。

◆それにしても「ROGUEスゴロク」は面白かった。ときどき#(シャープ)マークのサイズが違っているのもいい。だけど，3コマ目以前のネタがわからないのが悲しい。

大渕　正人（19）北海道

◆斎藤晋さんの「X68000現象を探る」はとても面白い。私の友人のX68000ユーザーを見ているとわかるのだが，X68000ユーザーはX68000に心底ホレ込んでいるので多少の借金は苦にならないようである。もしかして，3万人のユーザーのほとんどがそんなんだったりして……。

斎藤　肇（19）埼玉県

X68000ユーザーが借金してる人ばかりじゃないでしょうけど，いろんな意味でリッチなのは確か。なんとあのドラスピに至っては，12月現在で2万本を突破したという噂もあるし，いずれにしてもうらやましい限りです。

◆なんと懐かしい。「霜降り高原から」を読んで，X1の歴史を思い出してしまった。昔のX1の色はホワイト，ワインレッド，シルバーの3色だったですよね。そしてturboが出現してからオフィスグレー，ワインレッドとなり，turboIIからブラックの登場でした。このブラックモデル，最初は限定発売だったのが，いつからかブラックだけになってしまいました。そこで提案なのですが，次期X68000は「虹色のX68000」というコピーで，いろいろな色のものを発売するのです。マーブルなんていうのもいいかもしれません。シャープさん，ダメでしょうかこの話。

杉元　敏光（17）愛知県

X68000が発売された当初は，迷彩色だとか唐草模様もオプションでほしいという話があったように思うけど，確かにインテリアにマッチしたオリジナルカラーはぜひ希望したいところ。せめて次のマシンは，4色ぐらい最初から用意してくれてもいいはずですよね。

◆まあ，アレですね。その筋キーホルダーを所有する僕としてはですね，そう，なんというのかな，こうフツフツと血がたぎるワケですよ。現在の食文化から一歩踏み出した，それゆえいまの世の中に受け入れてもらえない「その筋缶詰セット」。こんなものもらったら，思わず神棚に3日間供えてですね，そうだなレビューでも書こうかな。あー，楽しみ。

田中　真一（17）三重県

◆プレゼントにあった「トドの大和煮」は私も食べたことがある。　福地　敏男（20）北海道

編集室でも熊の缶詰は食べた人がいるみたいだけど，トドはまだ誰も手を出していません。どんな味がするのかぜひ教えてくださいな。

◆12月号の「電脳遊戯民」を読んで，どうして日本のRPGが面白くないのか，その理由のひとつがわかったような気がする。ゲームの世界に入ってまでも枠にはめられるのが私は嫌いだったからなのだ。RPGではやりたくなくても強制的に善人にさせられて，親切な行為や人助けをしなければ先には進めない，というのが好きにはなれなかったからだ。

田島　啓一（23）福岡県

◆イースIIを解き終えたとき，「経験値稼ぎのRPGの時代は終わった」などと騒いでいた私でしたが，今度はM&Mをやってみると，「やっぱりRPGは経験値稼ぎだぜ」なんていっている私です。やはり大切なのは，そのゲームに合ったゲームデザインではないでしょうか。あまりセオリーにとらわれず，楽しければそれでいいのだと思います。　阿部　勝（16）秋田県

RPGだろうがAVGだろうが要はいかに楽しませてくれたか，だと思います。だから自由度が高くてさらに面白いシステムの完成，これが当面の課題だと思うんだけどね。

◆「真面目に遊ぶOh!X」は，とてもいいコピーだ



◀福原　徹（14）埼玉県
これは強力なメッセージ！　ところで，福原君も3万人のうちの1人だったりすんだよね。おっと，私らもそうか。（S. S.）



◀杉田みどり（17）宮崎県
同じ頑張れでも，こちらは女性読者から。感動だなあ。Oh!Xより「みどりさんのお便り」を心の励みにがんばっちゃいましょう。うん。

と思います。コピーに負けないようこれからも遊んでください。　　高橋　功（20）神奈川県

◆(で)さん，まだ埼京線があります。これこそ日本一です。　　田中　保史（18）埼玉県

関東近辺以外の方にはわかりづらいかもしれませんが，埼京線って新宿駅の貨物ホームを利用して貨車を使って乗客を運搬しているという，悲しい通勤快速なのです（本気にしないよーに！）。

◆「僕が大学を卒業するまでに，米子に情報産業がたくさん進出してくると嬉しいけど，結局，無理だろうなぁ」，と最近思っています。

森　弘（20）山口県

◆鹿児島大書籍部での出来事。「さすがにコンピュータ関係のコーナーはMS-DOSの本ばかりだなぁ」，「あれ？　なんでこんなところに音楽関係の本が……」。その本のタイトルは「THE MODS」。ロックグループの本をコンピュータコーナーに置いてしまうとは，鹿大生協はいい度胸してるぜ。　　塚本　雅俊（19）鹿児島県

◆友人たちと徹マンをするはずだったのが，な，なんと11月13日のAM 3：00～PM12：30まで，9時間半も「信長の野望・全国版」をX68000で遊んでしまったのだ。それも5人で。結果は徳川が天下統一。それにしても，エンディングはもっとドハデにやってよ，光栄さん。

松井　雅宏（17）京都府

徹マンに面子が5人。うーん，手慣れているなと思ったら，なんと17歳。コレコレ，大学に入ってからにしなさいって，こんな遊びは。それにしても信長を9時間とはギネスものかもね。

◆遅ればせながら，「殺意の接吻」を解き終わりました。それにしてもこのシリーズはエンディングが悲しいものばかりですね。せっかく終わっても，爽快な気分というより，夕日に向かってブラックコーヒーを飲みたくなるような，そんな気分になってしまいます（ワッハハハ，キザー）。でもテンポのよい音楽，見事なまでのグラフィック，そのどれをとっても満足のいくものでした。最後に甘えごとをひとつだけ。「部分的にでもいいから，コマンド入力もほしかった」。　　田中　五郎（20）奈良県

◆ドラスピマニアの方で「イマイチ縦横比が違うような……」，と思っている方は縦型画面モードにして，ディスプレイの垂直振幅の31kHzのツマミを右に回してみてください。あまりのリアルさに笑いが止まりません。これは源平やグラディウスにも有効です。でも遊び終わったあともとに戻すのがたいへんだけど，皆さん一度やってみてはどうでしょうか。

荘司　真吾（17）北海道

笑いが止まらないほどリアルになるんだったら，今度やってみよっと。

◆11月15日，僕の所属する自然科学部で毎年恒例の「千歳飴早なめ大会」が盛大に開催された。これは浪人中の先輩までエントリーしてしまうというビッグイベントであり，この僕は昨年1分30秒台を出して優勝したのだが，今年はそれ



◀宮本　康司（20）大阪府
ええっ？　初めてでこれだけ描ければたいしたものですよ。おまけにメッセージもワクまで凝ってるし。これからもがんばってよね。

より5秒もタイムを縮めたのにもかかわらず敗れてしまった。なんと1分6秒ですべてを飲み込んでしまった強者が現れたのだ。これにはちょっとかなわない。それにしても揃いも揃って高校生が千歳飴をバリバリやっている姿は，我ながらちょっと不気味です。

松本　篤志（18）東京都

◆我が武蔵工大の第59回文化祭（MIT祭）は，小泉今日子，太田貴子，そして新人の今井麻起子などのコンサートや，早飲み大会など盛況裏に終了し，協力スタッフとしては大いに喜んでいます。来年は創立60周年ということで，学校側からも企画の募集があり，私はX68000をホストにするパソコンネットの企画を提出しました。できればそのプログラムは私がぜひ作りたい。

武本　浩平（22）神奈川県

◆今年の文化祭での我が電気工学実験部の出し物は大ハズレであった。というわけで，「共通I/Oにつながって文化祭で客を呼べそうな物」という企画を掲載してください（自分勝手な奴）。

大戸　孝二（18）兵庫県

大戸君のために学祭ネタを2本ご用意してみました。それにしても松本君とこの千歳飴の早なめ大会っていうのは強力そう。

◆私はつい最近「おじさん」と呼ばれてしまいました。まだ19歳ですよ，どうしたらいいんでしょう。　　中山　剛（19）長崎県

大丈夫，いずれはみんなこのような経験はするんですから。私なんぞは，学生時代に某大手スーパーでバイトしたとき，おチビさんたちからさんざん「おじさん」呼ばわりされてからというもの，すべて悟ってしまい，こうして立派（？）に成長してきたのです。

◆最近はOh!Xが届くとまず見るのが「売ります，買います」のコーナーである。これでいいのでしょうか。　　林　一幸（32）滋賀県

猫は売るわ，セガのマシンが登場するわ，電子林まで，といったように最近のOh!Xのトレンドはこのコーナーにあるといっても決して過言ではないのです。ですから，遠慮しないでどしどし読みふけってください。

◆X68000を，編集室の方はいったいどのように呼んでいるのでしょうか？　いちいち「えっくすろくまんはっせん」と呼ぶのは面倒です。



◀松本　篤志（18）東京都
ちょっと古いネタだけど，まあいいか。ぜんまいちゃんのイラストはあぶない手合いが多いんだけど，これはクールにまとまってるもんね。

小比賀　亮介（18）宮城県

一般的には「ろくはち」と呼ばれています。

某メーカーのユーザーさんたちはなぜか「ぺけろく」と呼んでいるようですが，あれは邪道です。

◆先日，この私はナントあの『ファンロード』という雑誌を生まれて初めてこの肉眼で見てしまいました。本が置かれている位置も決して「主婦の友」や「少年ジャンプ」などのところにはなくって，それなりの場所に置いてありました。「うん，うん」とそれを見てひとり納得しながら，1メートルほど離れた場所から観察したあと，思わず手に取ろうとした瞬間，どこからか「あっ，ローディストだ」といわれてしまうのでないかと，勝手に恐怖におののいてしまうのでした。　　山口　明徳（18）埼玉県

これくらいで恐怖におののいていては，いまの時代は生き残れません。やはり『ファンロード』のネタとなって載ってしまう，これが一番恐ろしいパターンだといえるでしょう（おお，ついに禁句をしゃべってしまった）。

◆島田奈美を表紙に使えば，きっとOh!Xはいまよりもっと売れると思う。

中里　晋作（18）神奈川県

島田奈美もいいけど，これからは光GENJIを表紙に使って編集部のご近所にある武道館でコンサートのときに売る。やはり，これが一番強力ではないかと考えています。

◆「ヤングガン」を観てきました。とってもこの映画はよかったと思います。それで思ったんですが，これを機に西部劇が復活しないかなーと。僕としてはポール・ニューマンとレッドフォードのコンビで「明日に向かって撃て」みたいな，ちょっとモダンな感じのオシャレな映画が出てほしいと思います。別にポール・ニューマンが主演ならどんな映画でもいいんですがね。

安藤　淳二（19）京都府

アメリカの二世俳優たちって，親譲りの個性が光っているようでとてもいいですよね。それに比べて日本の二世さんたちときたらねえ……。

◆誰か知りませんか？　NTTのキャプテンがど



▲杉浦　豊　静岡県
いいところに気がつきましたね。やっぱり剣と魔法の世界じゃお姫様を助けなくちゃ。で，絵のもっていきかたもなかなかですよ。

こに行ってしまったのか。3年ほど前にあれだけ世間を騒がせたのに，いまではぜんぜん声も聞きません。いったいどこに消えてしまったのでしょうか。　　愛場　俊紀（28）北海道

いったいどこに行ってしまったんでしょうね，最近の日電のテレビCMにも出てこないようだし。

◆10月23日実施の河合塾「第2回早慶レベル模試」の小論文でなんと70点をとってしまい，しかも，全国版の学習解答例に載ってしまった。なぜこんな自慢めいたことを書いているかというと，これもOh!Xのおかげなのです。特に「知能機械概論」の有田さんのおかげなのです。早稲田の人間科学を目指している方は，これをもう一度読み直しておきましょう。ちなみに人工知能を人口知能と書いて，世間に大ハジをさらしてしまったのもこの私です。
畦地　真太郎（18）北海道

全国版の模範解答として掲載されたなんて凄いですね。これからもOh!Xのなかで活用できる部分があれば，受験生の方は大いにご利用ください。もしこれが成功すれば，近い将来Oh!X別冊として受験参考書『受験生の野望・全国版』が発売されるかもしれません（うちは学研じゃないってばっ）。

◆年内のF1もとうとう終わってしまいました。来年は中島がロータスに，鈴木がザクスピードに決まっています。ガンバレ中島，ガンバレ鈴木。ところで来年のコンストラクターズチャンピオンはいったいどこが取るのでしょうか。僕はやはりマクラーレンだと思うけど，ウイリアムズとマーチあたりも捨て難い。皆さんはいったいどこが取ると思いますか？
大倉　嘉宏（18）富山県

ノンターボ時代を迎え混戦が予想される次のF1。やはりマクラーレンが有利みたいだけど，通に聞くとベネトン・フォードあたりが穴馬的存在のようです。

◆PC-8001からいきなりX68000に買い換えたため，目が白黒じゃなくて65536色になってしまった。　　三好　正竜（24）神奈川県

◆我がX1turboがゲームマシンと化している，という現状を打破する道があった。それは「ビデオテロップ」である。先輩に頼まれて作ったやつの編集完成版を見た。凄かった，まるで映画みたい。デジタルテロッパーの威力は凄い。でも，おかげでビデオカメラもほしくなってしまった。　　長井　史朗（19）千葉県

やはりこれからはパソコン単体だけの機能ではなく，AV機器との融合も考えていく必要がありそうです。長井君みたいに，私はこんな使い方をしているよ，という方がいればまた教えてくださいね。

◆速報！　MZ-700でRPGの制作決定。拍手（パチパチパチー）。とりあえず期待していてください。受験生のくせに，現在だいたいのデータ設定は終わっています。合格したらいよいよプログラミングに入る予定。やっぱりMZ-700に不可能はない。　　田中　義章（14）北海道

パチパチパチー！　凄い，ぜひとも完成させて，といいたいところだけど，やっぱり受験が終わってからにしようね，田中君。

◆うちの高校の科学工学科にはMZ-700が2台あります。実習時間にこのMZを使っていたとき，S-BASICを返そうと別の教室に行ってみると，誰もいないその教室にはなんと1984年からのOh!MZが32冊も並んでいるでありせんか。僕はさっそく貸してもらおうと，慌ててこの本が誰のものであるか先生方に聞いて回りました。すると，これがまた僕の担任の先生のもので，貸してくれるように話をすると「もうあまり使わないから，お前に全部やる」といわれたのです。嬉しい＆やったぁー！
石川　俊太郎（18）大分県

ホント，石川君はいい担任の先生を持ったものです。

◆近年，「オタク」だとか，「オタッカー」なる言葉が巷で使われているようだが，古くより伝わるMZ-80K/Cシリーズの「おとりATTACER」こそが，本当の「オタッカー」ではないのか。
高橋　祐司（19）北海道

高橋君のメッセージはどうも意味不明だけど，祝一平氏の説によると，「オタク」には「オタク」，「オタッカー」，「オタッキー」ときて，その上にさらに「オタキスト」なる最上級の強者が存在するらしいのですが，これってホントなんですか？

◆つい先日，アメリカ合衆国国防省（ペンタゴン）がステルス戦闘機を公開しました。まあ，以前から機種不明機が墜落したとか，いろいろと話題にのぼっていましたが，そのコードナンバーについては，F-18ホーネットとF-20タイガーIIの間のナンバーが空いていることから，F-19ではないかといわれていたのが，ペンタゴンも意地になったのかどうかは知らないけれど，「F-117A」という名称に落ちついたようです。でも，そうなるとF-19はどうなってしまうんでしょう。　　小野寺　智宏（17）東京都

◆Oh!Xでは，テレ朝系の「ライブマン」という言葉を一度も見かけたことはないけど，この番組のなかで「X68000，X1turboII，MZ-2500，IWAI，SHARP」などの言葉が，コンピュータの画面を映したとき表示されていることを，Oh!Xの読者のどれくらいが知っているのだろうか。番組の内容もいいのに，幼児向け番組だからといって話題にならないのは，これを毎週ビデオに撮っている者としては少々さびしい。
川村　重夫（23）静岡県

別に幼児向け番組だからといって，除外しているわけじゃないけど，それにしても「IWAI」の文字だけはどう見ても怪しい。

◆最近，ようやくKamikazeを仕事のために購入しました。いざ使ってみると，職場で使用しているN5200のLANシリーズと比べて少々遅いような気がします。ただ，LANシリーズはLAN PLAN，LANFILE，LANGRAPHといった独立したソフトであることを考えると，簡単には比較できないとも思っています。これからこのKamikazeを使いこなしていきたいと思っています。
上原　俊夫（32）群馬県

X68000の実務ソフトに関してのご意見があまり届かないような気がしていましたが，上原さんのハガキを見てひと安心。実際に使っているユーザーの方からのレポートをたくさんお待ちしています。

◆四十路を越えても，恥も外聞もなくパソコンに熱中して，日ごろ女房に口うるさくののしられています。頭の中の脳味噌が固くなり，もはや自分からプログラムすることは不可能で，専用ソフトを買っては，秋葉原の電気街を喜ばせております。「信長の野望・全国版」などは，東



◀田村　憲生（19）広島県
うまいなあ。デザイン力がないとこういうのは描けませんものね。でも，文字と絵のアンバランスがやっぱり田村君だなあとホッとします。



◀森下　保　静岡県
来ましたよ。またしてもこの迫力。果たして彼の胸の内に秘められたものは意欲か，それとも復讐心か？　結局，同じことしか言えない私。

北・関東制覇のままストップしており，子供に笑われています。いまはワープロとして使うことに喜びを発見しています。

村川　三郎（42）埼玉県

◆若い者に負けじとがんばってソーサリアンに挑戦していますが，なかなかクリアできず毎晩悩んでいます。もう少し詳しい攻略法でも掲載してください。このごろは目が悪くなって，プログラムを打ち込むのもつらくなり，もっぱらゲームを楽しんでいます。

国分　大幸（55）福岡県

10代，20代の若い世代が多いなか，村川さんも国分さんも，高齢ながら（おっと失礼）がんばっていらっしゃるようで，頼もしい限りです。ワープロにしろゲームにしろ，楽しくパソコンを使っていただければそれだけで十分だと思います。これからもがんばってください。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集室では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。

仲　間

★「T・A・C」では，MZ-2500ユーザーの会員を募集します。現在，会員は30名。ディスク会報を毎月1回発行しています。入会ご希望の方は60円切手同封のうえ封書にて連絡を。〒569 大阪府高槻市大塚町2-55-4　北浦真一

★X1とS-OSを中心とした会報を発行する会を，現在運営しています。プログラムに興味のある方はたくさん参加してください。機種はマイナーな機種の方でも結構です。活動は2カ月に1回発行の会報を中心に行っています。会費は1回140～180円，入会金は100円です。参加ご希望の方は，小為替で100円を送ってくれれば会報のサンプルと案内書をお送りします。また，案内書だけの場合は60円切手同封のうえ封書でご連絡ください。〒811-42 福岡県遠賀郡岡垣町戸切794-3　筑紫高宏（21）

★X1/X68000ユーザーを対象とした「CZ club」では会員を募集します。入会にあたっての制限は特になく，技術レベルも問いません。活動は月1回発行の会報やPDSの配布などです。詳しくは60円切手5枚同封のうえ連絡を。折り返し会報をお送りします。〒458 愛知県名古屋市緑区ほら貝2-356 CZ club事務局　杉山寛夫（58）

★私たちコンピュータ大好き，ゲーム大好き，音楽大好きサークル「夢幻史」では，スタッフを募集します。X1シリーズを持っている方，X1をフルに使ってあげましょう。自分の楽しいパソコンの使い方をみんなで共有しましょう。興味のある方は60円切手同封のうえ封書にて連絡を。〒593 大阪府堺市宮園町25-609　柿内一宏(17)

★全国のS-OSファンの皆さん，ユーザーズクラブを作って互いに情報交換しませんか。当方で現在考えている活動内容は，オリジナルプログラムのディスクサービス，パソコン通信サービス(こちらは現在実施中)などです。積極的に参加したいと思われる方は住所，氏名，年齢，使用機種，メディアの種類を明記のうえ，簡単なお便りとともに170円分の切手を同封してご連絡ください。〒536 大阪府大阪市城東区鴫野東1-13-18　森喜一郎（23）

★「FUTURE FORUM」では，会報やパソコン通信を通じてAI(人工知能)などに関する情報交換を行っています。また，CP/MやMS-DOS，S-OSなどのOS上で作動するプログラムも紹介しています。詳しいことは120円分の切手を同封のうえ封書にて連絡を。〒192 東京都八王子市上野町28　沼野博胤

★「サークル・ラヴ」ではX68000と88シリーズを中心とした会員を募集します。ただいま会員数39名，活動は会報発行（X68000ユーザーにはディスク会報）を主に行っています。活動が軌道に乗ればBBSも考えています。詳しいことをお知りになりたい方は，ハガキに連絡先を明記のうえ連絡を。〒370 群馬県高崎市下小塙町610-9　田中克明

売ります

★プリンタMZ-1P17(新品，未開封)を2万7千円で。連絡はハガキで。〒343 埼玉県越谷市船渡632-15　山本光信（16）

★プリンタMZ-1P17にケーブル，リボンを付けて3万円で。連絡は往復ハガキで。〒305 茨城県つくば市観音台1-7-2　近森敬一（17）

★X1用データレコーダ CZ-8RL1(完動品，箱，付属品付き）を1万2千円前後で。連絡は往復ハガキで。〒731-02 広島県安佐北区阿部東3-23-1　藤原栄治（18）

★X1用データレコーダCZ-8RL1(完動品，箱，マニュアル付き）を送料込み1万円で。連絡は往復ハガキで。〒769-15 香川県三豊郡豊中町上高野2780-6　関敏和（17）

★プリンタCZ-8PK9(付属品，箱，1年間有効の保証書付き）を5万8千円で。連絡は電話番号明記のうえ往復ハガキで。〒457 愛知県名古屋市南区堤町2-16　高橋吉仁（18）

★FM音源ボードCZ-8BS1とプリンタCZ-8PC2，データレコーダCZ-8RL1にソフトを付けて4万5千円で(付属品すべて込み)。バラ売りも可。ご希望の方は希望価格明記のうえ往復ハガキで連絡を。〒506 岐阜県高山市上岡本町5-423　森雅樹

★ローランドDGのコンピュミュージックCMU-800，専用シンセCMU-810，シンクボックスCMU-802，MZ-2000用ソフト・インタフェイスをセットで5万円前後で。バラ売りも可。連絡は電話番号明記のうえハガキで。〒280 千葉県千葉市千城台西1-49-8　風間信幸（18）

買います

★X1用FDD・CZ-503F (I/F，ケーブル付き)を1万5千～2万円で。また，NEW Z-BASIC(CZ-8FB03増設RAM付き)を1万円前後で。連絡は往復ハガキで。〒651-11 兵庫県神戸市北区鈴蘭台南町4-5-6　七浦啓有（17）

★X1turboで使用可能なマウスと拡張I/Oボックスをそれぞれ送料込み2千円と8千円で。箱，保証書等不要。連絡は往復ハガキで。〒737 広島県呉市東畑1-10-11　高原浩（16）

★X1用FM音源ボードCZ-8BS1を1万円で。説明書，ソフトが付いていて完動品であれば傷，汚れあり箱なし可。〒959-39 新潟県岩船郡山北町大字府屋546-254　山口宏志（18）

★X1用カラーイメージボードCZ-8BV1またはCZ-8BV2を送料込み1万2千円で。連絡は付属品の有無を明記のうえ往復ハガキで。〒520-32 滋賀県甲賀郡甲西町夏見1903-44　野田創（18）

★カラーディスプレイTV・CZ-601DBを6万5千円で。連絡は傷，汚れ，状態，付属品の有無を明記のうえ往復ハガキで。〒051 北海道室蘭市新富町1-6-6　渡辺知巳（17）

★X68000ACE用1MB増設RAMボードCZ-6BE1Aを1万5千円で。連絡は往復ハガキで。〒289-13 千葉県山武郡成東町成東2470　安井忍（20）

★Oh!MZ1987年9月号に載っていたYAMAHAのYK-01かYK-20 を送料込み6千～8千円で。連絡は往復ハガキで。〒751 山口県下関市田倉御殿町1-9-5　竹本賢一（16）

★1200bpsモデム（＋通信ソフトなど）を1万円前後で。またX1で使えるドットプリンタを1万円前後で（ピン数問わず，24ドットであればCZ-8PC1を希望)。連絡は往復ハガキで。〒464 愛知県名古屋市千種区吹上1-4-12みのり荘て号室　橋居賢治（21）

バックナンバー

★Oh!MZ1986年6，7，9月号を送料込み各1,000円で。切り抜き不可，汚れ可。連絡は往復ハガキで。〒273-01 千葉県鎌ケ谷市南鎌ケ谷4-9-1　山崎博（18）

★Oh!MZ1987年2，9月号を送料込み各1,000円で。切り抜き不可。連絡は往復ハガキで。〒444-13 愛知県高浜市吉浜町北屋敷152-5　小笠原勝成（17）

★Oh!MZ1986年2，7，8，9月号を送料込み各1,000円で。傷，汚れ可，切り抜き不可。連絡は往復ハガキで。〒607 京都府京都市山科区四ノ宮泉水町5　笹倉岳臣（21）

DRIVE ON

このコーナーは，本誌年間モニタの方々のご意見を紹介しています。今月は1988年12月号の記事に関するレポートです。

●特集「心地よい雑音の話」での，乱数に対してフィルタをかけることで作曲する，というのはとても興味深いものでした。こういうやり方もあるのか，と驚きました。MusicBASICは，組曲「イース」の西川さんらしく，ほぼ完璧といえるMMLだと思います。相対ボリューム，アクセントなど，他のMMLでは見かけることがなくてもやっぱり必要な機能や，目立たないけどXコマンドやZコマンドなど便利そうな機能があって，細かい配慮が感じられます。フリーエリアがそのまま，ということも評価されるべきでしょう。グラフィック関係の命令が限られているのが残念といえば残念ですが。

今野　和浩（17）MZ-2500, PC-E200, PB-100, FX-780P　埼玉県

●特集のテーマである自動作曲についてだが，以前，「人間がいい曲と感じるのは心臓の鼓動くらいのテンポの曲が多い」と聞いたことがある。詳しいことはわからないが，そういった要素を取り入れてみるのも面白いと思う。たとえば，赤ちゃんは母親の心臓音を聞いて泣きやむが，それをコンピュータでシミュレートして音階をつければ，子守歌にできるかな，と考えたりする。

猪原　弘晃（18）XIC　兵庫県

●今回の特集を読んで，作曲はプログラミングと同じく，書き方さえわかればある程度のことができるようになるんだな，と思いました。作曲を難しく感じる最大の理由は，作曲の「作法」を知らないからなのですね。

中島　奨（22）MZ-2500, PC-9801VX　北海道

●Oh！Xは，S-OSなどを通して読者との関係がかなり深い情報誌なわけだ。パソコン誌といえば読者とメーカーとの接点となる重要な存在だ。その関係は一方通行であってはならない。そして読者の声はOh！Xだけでなくメーカーにも届かなくてはならない。読者は受け手であると同時に発信者であるべきだ。読むだけの本ではなく参加する本。それができるのはOh！Xを始めとする限られた雑誌だし，また，それがOh！Xの使命でもあると言える。

上野　壮也（17）MZ-1500　北海道

●「お買い得レポート・X68000現象を探る」は面白かったです。SOFTOUCH PRO-68Kなどで「X68000現象」という言葉が流行ったりして。僕としては，OSの「Humanity」がパソコンユーザーを呼んだ要因のひとつではないかと思うのですが。マウスだけでも操作できる（マウスがないと操作できない？）アイコン表示の画面が，これまでとは違った雰囲気と操作性を提供していました。それから，特集では和音などについてかなり詳しく取り上げていましたね。これを読んで，プログラマは単にプログラミングができるだけではダメだ，ということを見せられた思いがしました。

星　大地（15）MZ-700, PC-1475　静岡県

●メーカーには「使うほうの身になって考えて，もっと使えるソフトを出してくれ！」といいたい。またユーザーは，自分の使っているマシンのハードを知っておくべきだと思う。そのためにはメーカーはハードを公開するべきだし，パソコン雑誌があるのならそれらを読者にわかりやすく説明したりすることも大切だと思う。実際，雑誌に回路図などが載ったりすることもあるし，メーカーによっては頼めば回路図のコピーをくれたりする。要するにユーザーも，自分のマシンはどんなものか，ということに興味を持つべきだと思うわけだ。ときどき，BASICなどのシステムさえ立ち上げたことのない人間がいるらしいが本当だろうか。ゲームユーザーに終わってはいけないと思う。

松本　勝美（19）MZ-2200, XIマニアタイプ，XIturbo model 30　兵庫県

ごめんなさいのコーナー

1988年3月号　SLANG

P.124　式中で引数を持つ関数を使用すると正しくコンパイルできない場合がありました。以下のように訂正してください。

```
4419 CD A0 65
65A0 CD A1 4B
65A3 3E 00 32 67 63
65A8 C9
```

1988年12月号　MusicBASIC

P.113　3連符の処理や，小文字への対応が不十分などの不都合がありましたので訂正用プログラム（リスト1）を掲載いたします。使い方は，NEWON &HC000後，MMLをロード（CALL &HA8B0は絶対に行わないこと!!）して，訂正リストを入力後RUN。その後SAVEM“ファイルネーム”，&HA8B0，&HBFFF，&HA8B0でセーブしてください。

リスト1

```
10 'NEW MML DEBUG PROGRAM
11 POKE &HA913,107
20 POKE &HAB81,&H3C,&H47
30 FORI=&HAB8E TO &HAB97
40 POKE I,0
50 NEXT
60 POKE &HAD35,&HCD,&H58,&H47
70 POKE &HAD58,&HCD,&H7C,&H47
80 POKE &HADB0,&H58,&H30,&H13,&HCD,&HD1,&HAF
90 A$="CD D1 AF 03 3A 88 46 1E 2A ED 59 03 ED 79 03 C3 49 47 00"
100 MEM$(&HADCC,LEN(A$))=HEXCHR$(A$)
110 POKE &HB179,&HFE,&H8,&H30,&H15
120 FORI=&HB17D TO &HB184
130 POKE I,0
140 NEXT
150 RESTORE"DATA1":AD=&HB1AC:H=6:GOSUB 230
160 RESTORE"DATA2":AD=&HB35C:H=3:GOSUB 230
170 POKE &HB393,&HCD,&H1F,174
180 POKE &HB396,&HCD,&HED,175
190 POKE &HB3A3,&HCD,&H1F,174
200 POKE &HB3A6,&HCD,&HED,175
210 RESTORE"DATA3":AD=&HBDB7:H=5:GOSUB 230
220 END
230 FORI=1 TO H
240 READ A$:A$=HEXCHR$(A$)
250 MEM$(AD,LEN(A$))=A$:AD=AD+LEN(A$)
260 NEXT:RETURN
270 LABEL"DATA1"                                        :'SUM
280 DATA 2E 00 ED 78 03 FE 24 28 26 FE 30 38 20 FE 3A 30 :'5F4
290 DATA 1C 0B 16 03 ED 78 03 FE 30 38 12 FE 3A 30 0E D6 :'56C
300 DATA 30 67 7D 87 87 85 87 84 6F 15 20 E8 03 0B C9 16 :'62B
310 DATA 02 ED 78 03 FE 41 38 08 FE 47 30 04 D6 37 18 0A :'591
320 DATA FE 30 38 E9 FE 3A 30 E5 D6 30 67 7D 87 87 87 87 :'8A2
330 DATA 84 6F 15 20 DC 03 18 D5 00                      :'2F4
340 LABEL"DATA2"
350 DATA 7D 03 7E ED 79 23 FE 41 38 F7 FE 58 30 F3 7E FE :'8EA
360 DATA 2E 30 0A FE 23 38 06 03 04 ED A3 18 F1 C9 7D 08 :'5B5
370 DATA 03 CD 1F AE 55 08 C9 00 00 00 00 00 00 00 00 00 :'2C3
380 LABEL"DATA3"
390 DATA 1B 1B 1A 13 13 FE 24 CA 68 3E C3 AC 3F 7E FE 7D :'6AF
400 DATA 28 06 04 ED A3 03 18 F5 D1 C3 EB 3E 1A FE 7D 28 :'74C
410 DATA 17 FE 61 38 06 FE 7B 30 02 D6 20 12 13 7B D6 CF :'69A
420 DATA 7A DE AF D2 0C AB 18 E4 ED 5B 8C 46 21 00 00 C9 :'790
430 DATA FE 02 DA 0C AB 32 6D 46 C9                      :'43F
```

バグに関するお問い合わせは

☎03(263)2230（直通）

月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

行く手になにが待ちうける？'80年代最後の年

▼久し振りのマシン語大特集はいかがでしたか。ひとくちにマシン語入門といってもいろいろなアプローチの仕方が考えられるものですね。Z80と68000という違いはあっても，マシンがよりパーソナルであるために必要なもののひとつがマシン語だ，ということは同じだと思います。

加えて全機種共通システムでは，待望のニューエディタアセンブラ，REDAが無事発表の運びとなりました。S-OSをフルに活用するためにも，「究極の言語」アセンブラに親しむためにも，皆さんぜひがんばって打ち込んでください。

▼「爆発，そして完成へ」。Z80マシン語ゲーム工房が今月で最終回を迎えます。応援ありがとうございました。7回の連載に対する意見・感想や，今後，筆者の村田敏幸氏に期待すること，また次のマシン語連載はどんなものがいいか，などについてご意見をお寄せください。

▼筆者のスケジュールの都合により長らく休載させていただいた連載「Between The Lines」ですが，いよいよ来月から復帰します。勝本信氏の歯切れのいいエッセイが，今度はどんな話題を提供してくれるか，おおいに期待できますね。

▼さて，4月号では1988年度GAME OF THE YEARが発表されますが，ただ今ぞくぞくと推薦するゲームソフトへの投票が届いています。応募締切は2月15日。まだの方，お早めにどうぞ。あなたの1票がGAME OF THE YEARの，そして1989年のゲーム界の行方を決定するのです。

▼年末年始の休みはいかが過ごされましたか。え？　Oh！Xの1月号を見ながら電子サイコロを作ったりゲームやって遊んだりしていましたって？　ずいぶん充実した日々だったようですね。編集室には，皆さんからのクリスマスカードや年賀状がたくさん届きました。華やかな色遣いのカードもすごいけど手のこんだモノクロ作品もしぶい。力作をどうもありがとう。いずれ紹介しますからお楽しみに。それから受験生の皆さん，追い込みですね。季節がら，風邪などひかぬよう気をつけてください。健闘を祈ってます。

投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討の上，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒102 東京都千代田区九段南2-3-26井関ビル
日本ソフトバンク出版部
Oh！X「テーマ名」係

SHIFT・BREAK

▶編集室にあのM1（KORGのミュージックワークステーション）があった。私もM1を持っているけれど，出荷時に入っているサンプル曲が違っていて，私のよりかっこよくなっていた!!　このM1編集室のものなんですか？　えっ，（U）氏のですって！　こ，これはうーまーいーぞー。
最近の味っ子はパワーアップしたと思う（善）

▶近頃は中学生のX68000ユーザーがいるそうな。ああ，どんどん時代に取り残されてゆくXICsユーザーの私。しかしこの前（とは言っても去年の話だが），私はMZ-1500ユーザーの女性に会ってしまったのだ。彼女はMZ-700用スペハリを打ち込むヒマがないと嘆いていたっけ。その時私はMZユーザーの底力を見たような気がした。（R.K.）

▶UFOキャッチャーを皆さん知っているだろうか。ある日，ヌイグルミに混じって赤のパンティが入っていたので友達が冗談半分でやったら取れてしまった。新宿西口のゲームセンターです。（H.K.）
（せっかく，一緒に編集後記を書こうってさそったのに一人でそういう話をしちゃうんだもんなー。欲求不満の○○生にはついていけない……（で））

▶今月はなぜか書くことがない。で，思いつくまま。スキーに行きたいけど金と時間が……。マイク＆ザ・メカニックスの新譜はいいぞ。クリスマスどうしようかなぁ，冬は北風が身にしみるなあ。あ～あ，今年は早い1年だった（もう去年だって）。あ，そうそう明けましておめでとうございます。
（一応自粛のふりをするC.W.）

▶バグは怖いが，バグが出ないプログラムはもっと怖い。ポコポコとコーディングして走らせた数千行のマシン語プログラムが動いてしまったりすると凄く慌てる。いろいろ意地悪な条件で試しても，他人に使ってもらってもバグが発見されないときは，一から作り直そうという気になる。けど，現実にはそのまま使っちゃうんだよね。あー怖い怖い。（Mu）

▶「個人」は好きだが，その集合体である「組織」は嫌いである。「組織」の中にいると「早くみんなと同じになり。そうすれば楽になるよ」と囁く声が聞こえる。それと闘うだけで精神は疲労する。人と人との関係はすべて相対的なものではあるが，他人との差別化によってしかアイデンティティが得られないのだとしたら，それはきっと毒されている。（K）

▶発売日に遅れること2日。「めぞん一刻完結編スペシャル」をやっと手に入れることができた。響子さんがしゃべる，グラフィックを描き起こす，ということで大いに期待していたのだが，グラフィックの色数は少ないし，画面のレイアウトはおかしいし，手抜きが目立つ。出せば必ず買ってしまうというファン心理を甘くみないでくださいね。（KO）

▶日本の地価を合計すると，アメリカ合衆国4コ分になるそうです。おかしいなぁ。どーしてそんなことがありえるんだろう。で，面積は25倍だから，ざっと計算すると日本の地価はアメリカの100倍だそうです。この国って，そんなに価値のある所だったっけ？？？（今のところ）徴兵制が無いのは有難いけど。ま，今年は良い年でありますように。（M）

▶kとg，tとd，sとzなどのように，発音方法がある1点だけ異なる組み合わせをミニマルペアというそうだが，これらの音を含む日本語のカナは，かとが，たとだ，しとじ，などのようにビジュアルな意味でもミニマルペアだ。なぜか知らないが，おかげで最近ワープロ入力しながらしょっ中ミニマルミステイクしてしまう。これって年がじら。（よ）

▶編集者4人の机には原稿整理用にX68000が3台，X1turboが2台，PC-9801が1台，プリンタが3台（むろんマシン室は別）。うむ，ちょっと目には某所より恵まれているような気がするぞ。Musicstudio用のデータはなんで同時演奏が8トラック以下なのに，MIDIチャンネルの10とかを使うんだ？　世の中MT-32ばかりじゃないぞ。（M1を買ったU）

▶調子の悪いカットシートフィーダに，ゼムクリップを力任せに押し込んでステーを固定してやったらこれまで以上に調子がよくなった。そういえばディスプレイにゆらぎが生じたときは，平気で外側をポコポコ手で叩いていたような気もする。ハードがいくら進歩しても，人間側の対応はさほど変わっていない。このアンバランスはなんとなく愉快。（N）

▶昔のマシン語特集といえば，必ずモニタコマンドの使い方から始まったもの。そして多くの人がアセンブラを持たずにマシン語に入門したのでした。それを思えば，初めてのマシンがX68000という人には「マシン語」という言葉に対するイメージからして違うかもしれません。だってX68000のシステムにはマシン語モニタなんて付いてないんですよね。（T）

microOdyssey

ミニコンポが壊れ，アンプとＣＤプレイヤーを買い替えた。そこで私は，７つのリモコンとの関係を清算することにしたのだが……。

リモコンが付いているのは大小２つのテレビ，CD，LD，VHS，Beta，そしてAVアンプだ。これにカセットやチューナを買い替えるとどうなっちゃうんだろう（まるで貧乏人のリモコン沢山じゃないの)。幸い，アンプのリモコンには学習機能があるので，ビデオやＣＤの操作も覚えさせることができた。だが，これは便利と思いきや，気分は一向にカウチポテトになってこない。多機能なリモコンは大きく重いこと，小さなボタンがたくさん並んでいるために目的のボタンを探り当てるのに時間がかかることなどが原因だ。結局，状況に応じて２つばかりのリモコンを選択し手元に置くという方法に落ち着いてしまった。ちなみに録音ボタンについては私は学習させないことにした。

映像に対してものぐさな態度をとるためにはリモコンの存在は重要だ。ソファーでくつろぐカウチ族も，茶の間でまるくなるコタツネコも，お行儀の悪さは変わらない。リモコンは手探りでつかみ，無意識にホームポジションをキープする。ときにミカンの汁が顔にはねたりしなければ画面から目が離れることはない。もちろんテレビに集中というわけではなく，些細なことで視線を動かすような能動的な行為はものぐさの精神に反するということだ。

では，こうした理想的環境（どこが？）のためにリモコンはどうあるべきか。たとえば，ビデオのリモコンでは，▶（再生），■(停止)，❚❚（一時停止），▶▶（早送り），◀◀（巻き戻し）などのボタンが整然と並んでいる。特に疑問に思うのは■と❚❚が並んで位置しているリモコンがほとんどであることだ。ストップとポーズはよく似た動作をするが，使い方から考えれば，まったく違う機能である。▶や■を押すときは目で確認するだけの余裕がある。それに対して❚❚は，▶▶，◀◀，そしてコマ送りなどと同様，映像を見るという連続的な行為のなかで使用されるものであり，指先の感覚だけで操作したいボタンだろう。こうした一連の操作には１つひとつ分離した四角いボタンよりファミコンの✚ボタンのほうがよほど優れている。これと❚❚があれば，だいたいの操作が手探りで可能になる。

さらに言えば，早送りの ▶▶ マークを見て，どうしてダブルクリックを思いつかないのだろう。一般に再生中に▶▶を押すと早送りとなるが，先のほうまで飛ばすためには一度ストップボタンを押してから操作する必要がある。これは機械側の都合でしかない。▶▶をダブルクリックすると自動的に停止し，高速巻き上げがかかるようにすればよいのではないだろうか。

リモコンのボタン配置は，明らかに視覚的なわかりやすさを重視したものだ。そのほうがユーザーに操作方法を理解させるには都合がよいからだろう。マニュアルだってきちっと図解すれば誰も文句は言わない。だが，理解しやすいものが使いやすいものとは限らない。このからくりに気づいたとき，身の回りにあるすべての操作体系に疑いをもつようになってしまった。家電製品と比べてパソコンやワープロの操作性の悪さ，マニュアルのわかりにくさを指摘する人は多いが，本当にそうなのだろうか。（Ｔ）

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(233)3312
	〃	書泉ブックマートBI	03(294)0011
	〃	書泉グランデ5F	03(295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(257)2660
	八重洲	八重洲ブックセンター3F	03(281)1811
	新宿	紀伊国屋書店本店	03(354)0131
	高田馬場	未来堂書店	03(200)9185
	渋谷	大盛堂書店	03(463)0511
	池袋	西武百貨店11Fブックセンター	03(981)0111
	〃	西武百貨店9F コンピュータ・フォーラム	03(981)0111
	町田	久美堂東急ハンズ店	0427(28)2783
神奈川	横浜	有隣堂横浜駅西口店	045(311)6265
	〃	有隣堂ルミネ店	045(453)0811
	藤沢	有隣堂藤沢店	0466(26)1411
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ５	0471(64)8551
	船橋	西武百貨店10Fブックセンター	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	0472(24)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Ｘの定期購読をご希望の方は，最寄りの郵便局にある払込用紙に，

口座番号　東京1-29307
加入者名　株式会社日本ソフトバンク

とご記入のうえ，**年間購読料6,500円**を添えてお申し込みください。その際，裏面の通信欄に「○年○月号よりOh!Ｘ定期購読希望」と忘れずに明記してください。なお，すでに定期購読をご利用いただいている方には，購読期限終了と同時にご通知申し上げますので，同封の払込用紙をご利用ください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS（株）にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(238)0700

Oh!X　2月号

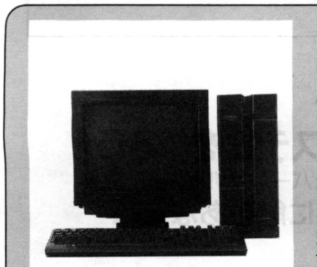
■1989年２月１日発行　定価540円　■発行人　孫　正義　■編集人　笹口幸男
■発行元　(株)日本ソフトバンク
■出版事業部　〒102 東京都千代田区九段南2-3-26 井関ビル　☎03(261)4095　FAX 03(262)8397
編集室☎03(239)4156
出版営業☎03(261)4095
広告営業☎03(297)0181
■本　社　〒102 東京都千代田区九段南2-3-14　靖国九段南ビル　☎03(263)3690㈹
TELEX 東京　232-4614JSBTYJ　FAX 03(263)3660
■西日本営業部　〒541 大阪府大阪市東区南本町2-6　明治生命堺筋本町ビル10F
☎06(264)1471㈹　FAX 06(264)1481
■印　刷　凸版印刷株式会社

信用と実績を誇る BASIC HOUSE

北関東最大の68000専門店

SHARP SONY NEC YHP 横河ヒューレットパッカード Apple Computer

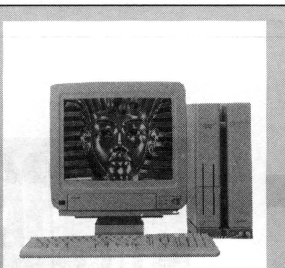
BASIC HOUSEおすすめ特別セット



Hyper COBURA SET A

CZ-601C	¥319,800
CZ-603D	¥ 84,800
ハンデープリンタ	¥ 24,800
メロディBOX	¥ 16,800
当社オリジナルソフト(7本)	¥ 74,400
5インチ2HD10枚PDSソフトサービス	
定価合計	¥520,600
超特価	¥389,000 〒1,000

長期クレジット 頭金9,000 12,800×36回



Hyper COBURA SET B

CZ-611C	¥399,800
CZ-603D	¥ 84,800
ハンディープリンタ	¥ 24,800
メロディーBOX	¥ 16,800
当社オリジナルソフト(7本)	¥ 74,400
5インチ2HD10枚PDSソフトサービス	
定価合計	¥600,600
超特価	¥459,000 〒1,000

長期クレジット 頭金9,000 15,200×36回

BASIC HOUSEオリジナル

X68000シリーズ

B6-6301	BASIC拡張関数パッケージ	¥ 9,800
B6-6302	CP/Mエミュレータ	¥19,800
B6-6303	アイコンエディタ	¥ 4,800
B6-6304	ディスクキャッシャー	¥ 6,800
B6-6305	C言語ライブラリ	¥ 6,800
B6-6306	BASIC拡張関数パッケージC言語ライブラリ付	¥14,800
B6-6307	TOYS & TOOLS	¥ 6,800
HANDY PRINT jack		¥24,800
MELODY BOX MIDIインターフェース		¥16,800
KGB-X68ADC	16ch12ビットA/D変換ボード	¥128,000
KGB-X68PIO	アイソレーション16Bit入出力ボード	¥68,000
KGB-X68UNB	ユニバーサルボード	¥ 6,800

MZシリーズ

B7-2501	PC-8801→MZ-2500テキストコンバータ	¥ 3,000
B7-2502	PC-8001→ 〃	¥ 3,000
B7-2503	PC-6001→ 〃	¥ 3,000
B7/2504	FM-77 → 〃	¥ 3,000
B7-2505	MSX → 〃	¥ 3,000
B7-2506	S1/L3 → 〃	¥ 3,000
KGB-MZ1	超低価格計測制御ボード	¥15,500

X1/X1turboシリーズ

KGB-X1S	低価格アナログデジタル入出力ボード	¥19,800
KGB-HDIF	X1turbo専用ハードディスクインターフェースボード	¥16,000
KGB-PIO	高級絶縁型パラレル入出力ボード	¥42,000

KGB-AD12	高級16ch 12BitA/D変換ボード	¥118,000
KGB-DA4	高級4ch 12BitD/A変換ボード	¥98,000
KGB-488	GP-IBインターフェース(マニュアルソフト付)	¥58,000
B6-3301	PC98↔X1turbo相互ファイルコンバータ	¥ 4,800

PC-8801/PC-9801シリーズ

KGB-PC1	KGB-MZ1のPC-8801版	¥15,500
KGB-98S	PC-9801シリーズアナログ入出力	¥19,800
	デジタル入出力ボード(D/A付)	¥25,000

Macintoshシリーズ

PRINT jack	プリンタドライバー	¥38,000
MOUSE jack	マウスドライバー	¥ 4,800
MELODY BOX	MIDIインターフェース	¥19,800

X68000 SOFT HARDプライスリスト

BASIC HOUSE特価CALL

ハードウエア

CZ-6BE1 1MB内蔵RAM(CZ-600用)	¥35,000
CZ-6BE1A 1MB内蔵RAM(ACE用)	¥28,000
CZ-8NS1 カラーイメージスキャナー	¥188,000
CZ-6BN1 スキャナーパラレルI/F	¥29,800
CZ-6BC1 FAXボード	¥79,800
CZ-6BP1 数値演算プロセッサーボード	¥79,800
CZ-6BV1 ユニバーサルI/Oボード	¥39,800
CZ-6BG1 GP-IBボード	¥59,800
CZ-8NT1 トラックボール	¥13,800
CZ-6BF1 RS-232Cボード(増設用)	¥49,800
CZ-6EB1 拡張I/Oボックス	¥88,000
KGB-X68 UNB ユニバーサルボード	¥ 6,800
KGB-X68PID アイソレーションI/Oボード	¥68,000
KGB-X68ADC 16ch12BitA/D変換ボード	¥128,000
KGU-HDPR ハンディープリンタ	¥24,800
KGU-X68MD MIDIインターフェース	¥19,800
CZ-6BM1 MIDIボード	¥26,800

ソフトウエア

CZ-212BS ビジネスプロPRO-68K	¥68,000
CZ-220BS DATA PRO-68K	¥58,000
CZ-226BS CARD PRO-68K	¥29,800
CZ-214MS SOUND PRO-68K	¥15,800
CZ-213MS MUSIC PRO-68K	¥18,800
CZ-215MS Sampling PRO-68K	¥17,800
CZ-221HS NEWprint shop	¥19,800
CZ-223CS Communlcation PRO-68K	¥19,800
CZ-211LS C Compller PRO-68K	¥39,800
EW 日本語ワープロ	¥38,000
ビジレスAD68K 表集計データベース	¥98,000
KamiKaze 統合化ソフト	¥68,000
C-TRACE グラフィックツール	¥68,000
Z'S STAFF PRO-68K 〃	¥58,000
Hyper UD 〃	¥16,800
X Link PRO-68K 通信ソフト	¥19,800
CP/M-68K エミュレータ	¥19,800
CZ-237MS music Studio PRO-68K	¥25,800

ゲーム

ハウメニロボット	¥ 9,500
魔神宮	¥ 7,800
グランドマスター	¥ 9,800
DOME	¥ 9,800
上 海	¥ 6,500
スペースハリア	¥ 6,800
T.D.F	¥ 6,800
源平討魔伝	¥ 7,800
ザ・ラスベガス	¥ 9,800
レリクス	¥ 7,200
マンハッタンレクイエム	¥ 7,800
桃太郎伝説	¥ 7,800
ツインビー	¥ 7,800
アルカノイド	¥ 7,800
沙羅曼蛇	¥ 8,800
フルスロットル	¥ 8,800
熱血高校ドッジボール部	¥ 7,800
A列車で行こうII	¥12,800
ザ・リターンオブイシター	¥ 7,800

新春お年玉 大特価

少数限定

(X1G MODEL/30)

● CZ-822CB	標準価格	¥118,000 →	¥44,800
820DB	〃	¥ 79,800 →	¥35,000
			¥79,800

(X1twin)

● CZ-830CBK	標準価格	¥ 99,800 →	¥64,800
820DB	〃	¥ 79,800 →	¥35,000
			¥99,800
● MZ-1P17/1P17B			¥26,500

社員募集

職　種：ソフトウエア技術者
ハードウエア技術者
セールスエンジニア
マイコンショップ店長候補

資　格：35歳位までの男女経験者

給　与：能力、経験、年令を考慮し優遇

待　遇：昇給年1回、賞与年2回
各種保険年金制度、海外研修制度あり

休　日：日、祝祭日、年末年始、夏季

勤 務 地：本社及び各事業所

応募方法：履歴書をご送付ください。
後日面接日を通知いたします。
応募の秘密は厳守します。

株式会社計測技研　本社営業部／マイコンショップ／通販部　宇都宮市竹林町503―1　TEL0286-22-9811　FAX0286-25-3970

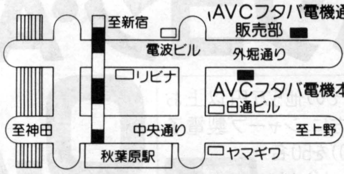
マイコンショップ BASIC HOUSE　お申し込み・お問い合せは ☎0286-22-9811(代)

秋葉原価格でローンができます
電気の街秋葉原で
24年の信用!!

AVCフタバ

AUDIO・VIDEO・COMPUTER

☎03(253)7661



AVCフタバ電機
〒101 東京都千代田区外神田2-9-8
神田ユニオンビル ☎03-253-7661(代)

今すぐ もよりの電話から	仙　台 022-264-3704	名古屋 052-452-3271	広　島 082-295-6873
札　幌 011-611-5104	新　潟 0252-75-4175	大　阪 06-311-3931	福　岡 092-481-2494

X68000の情報のすべて!(当店はX68000の認定代理店です。お気軽にご相談下さい)

CZ-8PC2 標準価格¥69,800 (今、プリンターを探してる人へフタバのお買得情報。)

C.G.やカラーイメージボードで取り込んだ映像を7色で鮮やかにハードコピー。JIS第1水準・第2水準の漢字をはじめ、多彩な文字種が使え、文書作成には24×24ドットの高品位で対応。

●オプション黒色リボンカセット　CZ-8PC-1 ¥700
　　　　　カラーリボンカセット　CZ-8PC-2 ¥800

※信号ケーブルおよび黒色/カラーリボンカセット各1個付

●CZ-8PC2の主な仕様

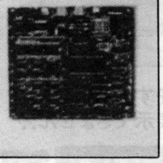
印字方式	ドットマトリクス・ノンインパクト熱転写
文字種類	288種〔英数・記号・カナ(またはひらがな)・その他〕
文字ドット構成	19(タテ)×15(ヨコ)ドット〔パイカ文字、エリート文字〕、19(タテ)×10(ヨコ)ドット〔縮小文字〕 24(タテ)×24(ヨコ)ドット〔漢字〕、24(タテ)×12(ヨコ)ドット〔半角文字〕
印字速度	45字/秒〔パイカ文字〕、30字/秒〔漢字〕
電源・消費電力	AC100V、50/60Hz、40W(印字動作中)、15W(待機中)
外形寸法・重量	幅365×奥行253×高さ65(mm)、約3.5kg



特価¥4?,800

お支払例 ¥8,168×6回 ¥5,038×10回
¥4,236×12回 ¥3,450×15回

CZ-6BC1



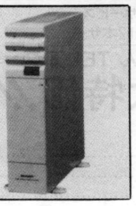
FAXボード。拡張I/Oスロットに装着し電話回線を利用してデーター通信を行う事ができる。

標準価格……¥79,800

特価　¥6?,000

お支払例 ¥5,920×12回 ¥4,071×18回
¥3,712×20回 ¥3,147×24回

CZ-6TU



RGBシステムチューナカラーディスプレイで、テレビ番組が楽しめます(200ラインアナログRGB)、ビデオ入力端子付。

¥35,800

特価　¥28,800

現金一括払

FDD搭載タイプの場合

CZ-601C	¥319,800
CZ-611D	¥145,000
合　計	¥464,800

激安

CZ-601C	¥319,800
CZ-601D	¥119,800
合　計	¥439,600

何故か安い

CZ-601C	¥319,800
CZ-603D	¥84,800
合　計	¥404,600

一番安い

(こんな表示で申し訳ない!はっきり書くとおこられます。)

(注)CZ-611Dは0.31mmと、グラフィックをされる方にはすぐれたモニターです。CZ-601DはTVも写る、皆様に人気のあるモニターです。CZ-603DはTVは写りません、パソコンのディスプレイとしてご使用下さい。

もちろん、分割払いもできます。

実装密度をさらに追求、信頼性を高めたハイコストパフォーマンスFDモデル。HDモデル。



※写真はCZ-601C-BK+CZ-603D-BK

■*本体+キーボード+マウス・トラックボール* ***CZ-601C-GY***(グレー)・***-BK***(ブラック)*標準価格319,800円*

■*14型カラーディスプレイ****CZ-603D-GY***(グレー)・***-BK***(ブラック)*標準価格84,800円(チルトスタンド同梱)*

HDD搭載タイプの場合

CZ-611C	¥399,800
CZ-611D	¥145,000
合　計	¥544,800

バカに安い

CZ-611C	¥399,800
CZ-601D	¥119,800
合　計	¥519,600

群を抜いて安い

CZ-611C	¥399,800
CZ-603D	¥84,800
合　計	¥484,600

どうして安い

(申し訳ないが値段は言えない程安いんです。)
分割の価格はどうぞお気軽にお問合せ下さい。

初めて買われる方へ。

HDD搭載タイプと云うのは20MBのハードディスクが付いています。
とりあえず初心者には必要ないだろうと思われますので601C+601Dをお勧めします。又他のオプションと併売する場合は充分な値引も考慮します。

型番	品名	標準価格	販売価格	お支払例
CU-14BD	ディスプレイ	¥64,800	¥?7,800	¥3,601×15回
CU-14ED	ディスプレイ	¥79,800	¥?2,600	¥3,346×18回
CU-14AD	ディスプレイ	¥84,800	¥?3,800	¥3,422×18回
CU-21CD	ディスプレイ	¥139,800	¥1?0,000	¥5,408×24回
CZ-820D	ディスプレイ	¥79,800	¥?4,800	¥3,375×15回
CZ-880D	ディスプレイ	¥109,800	¥8?,000	¥4,081×24回
CZ-603D	ディスプレイ	¥84,800	¥?3,800	¥3,137×24回
CZ-601D	ディスプレイ	¥119,800	¥??,000	¥3,203×36回
CZ-611D	ディスプレイ	¥145,000	¥1?3,000	¥3,892×36回
BF-68PRO	CRTフィルター	¥19,800	¥1?,800	現金一括払
CZ-502F	FDD(2DD)	¥99,800	¥7?,000	¥3,172×30回
CZ-503F	FDD(2D)	¥49,800	¥3?,800	¥3,219×12回
CZ-6BE1A	1MB 増設RAMボード	¥38,000	¥3?,800	¥3,388×10回
CZ-6BE2	2MB 増設RAMボード	¥79,800	¥6?,000	¥4,071×18回
CZ-6BE4	4MB 増設RAMボード	¥138,000	¥1?8,000	¥3,720×36回

型番	品名	標準価格	販売価格	お支払例
AN-8TU	RGBシステムチューナ	¥35,800	¥28,800	現金一括払
CZ-8PK7	プリンタ(80桁)	¥122,000	¥9?,000	¥3,238×36回
CZ-8PK8	プリンタ(136桁)	¥152,000	¥1?7,000	¥3,169×48回
CZ-8PK9	プリンタ(80桁)	¥89,800	¥?0,000	¥3,442×24回
CZ-6VT1	カラーイメージユニット	¥69,800	¥5?,000	¥3,562×18回
CZ-8BV2	カラーイメージボード	¥39,800	¥31,800	¥3,498×10回
AN-160SP	アンプ内蔵スピーカー	¥59,800	¥?8,000	¥3,053×18回
CZ-8BS1	FM音源ボード	¥23,800	¥19,800	現金一括払
CZ-6BN1	スキャナ用パラレルボード	¥29,800	¥24,000	現金一括払
CZ-6BU1	ユニバーサル10ボード	¥39,800	¥3?,000	¥3,520×10回
CZ-6BG1	GP-IBボード	¥59,800	¥4?,000	¥3,053×18回
CZ-8TM1	モデム	¥29,800	¥25,000	現金一括払
CZ-8TM2	モデム	¥49,800	¥?9,000	¥3,608×12回
CZ-8NT1	トラックボール	¥13,800	¥12,500	現金一括払
CZ-6SD1	システムラック	¥44,800	¥3?,800	¥3,312×12回

型番	品名	標準価格	販売価格	お支払例
CZ-6BF1	増設RS232Cボード	¥49,800	¥4?,000	¥3,013×15回
CZ-6BP1	数値プロセッサボード	¥79,800	¥6?,000	¥3,147×24回
CZ-6EB1	I/Oボックス	¥88,000	¥7?,000	¥3,442×24回
【OS9/X68K近日発売――予約販売開始!先進のパソコンに応える先進のオペレーティングシステム】				
CZ-227BS	TOP財務会計	¥200,000	¥1?8,000	¥4,279×48回
CZ-213MS	MUSIC PRO-68K	¥18,800	¥15,800	現金一括払
CZ-214MS	SOUND PRO-68K	¥15,800	¥13,800	現金一括払
CZ-212BS	ビジネス PRO-68K	¥68,000	¥5?,000	¥3,435×18回
CZ-211LS	Cコンパイラ PRO-68K	¥39,800	¥3?,000	¥3,520×10回
CZ-141SF	NEW-Z BASIC	¥18,800	¥15,800	現金一括払
CZ-137SF	turbo Z's STAFF	¥19,800	¥16,800	現金一括払
CZ-133SF	モデムターミナルソフト	¥25,800	¥2?,000	現金一括払
	Z'STAFF PRO-68K	¥58,000	¥47,000	¥3,541×15回
	kamikaze	¥68,000	¥5?,000	¥3,499×18回

X1turboZ



NEW-Z BASICは後で買えばいい。ハイグレードモニタをセットして驚異の価格。

CZ-880C……¥218,000
CZ-880D……¥109,800
合計………¥327,800

特価　¥1?3,000

お支払例 ¥16,003×12回 ¥8,506×24回
¥5,959×36回 ¥4,685×48回

X1turboZII



X1turboZの本格派セット。TV付2モードオートスキャンディスプレイ。

CZ-881C……¥179,800
CZ-880D……¥109,800
合計………¥289,600

特価　¥2?4,000

お支払例 ¥20,720×12回 ¥11,013×24回
¥7,716×36回 ¥6,067×48回

X1twin



HEシステムを搭載、最上級ゲーム機とパソコンが合体。

CZ-830C……¥99,800
CZ-820C……¥79,800
合計………¥179,600

特価　¥94,800

お支払例 ¥8,769×12回 ¥6,030×18回
¥4,661×24回 ¥3,265×36回

X1turboZIII



X1ターボシリーズの独自の機能を全継承VCCIゼロdB基準に適合させた。

CZ-888C……¥169,800
CZ-860D……¥99,800
合計………¥269,600

特価　???

応談　価格はご相談に応じます、電話でお問い合せ下さい。

●分割回数は3回~48回まで自由に選べます。

●セットの組合せは自由!広告に出ていない他の機種はお問合せ下さい。

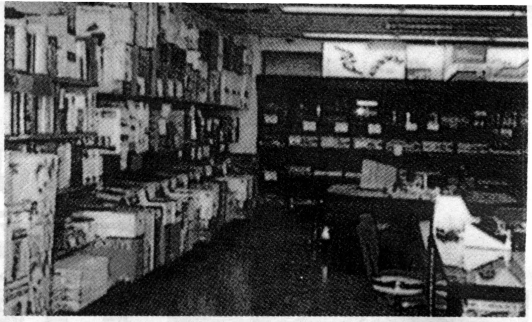
●**頭金なし**(手軽な電話クレジット) ●**製品先取り**(お支払いは約1~2ヶ月後から) ●**低金利クレジット**(1回の支払いは2,700円以上で3~48回。ボーナス併用も可) ●**カレッジクレジット**(保証人なし。但し満20歳以上の学生の方) ●**18歳未満の方**(ご両親が代理購入者としてお申し込み下さい) ●**納期**(通常の場合、当社に申込書が到着後1週間以内。特に人気のある商品で品薄の場合、少々納期が遅れることがありますので御了承下さい) ●**完全保証**(すべてメーカー保証書付。アフターケア万全) ●**全国代引**(お届けした者に、代金をお支払いいただく方法です。但し手数料1,000円)

AM10時からPM9時まで受付 日曜・祝日も営業

パソコン・AV専門
O.A.ランド
新年大特価セール

本体・CRT・その他5万円以上お買い上げの方に、シャープ製電子メモ(PA-370)を50名様に新春プレゼントします。お早目に！





セール期間
◀'89 1・16➔2・15

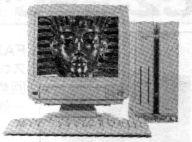
ランド特選!! SHARP X68000 ACE-HD / ACE セット

ACE-HD セット
- CZ-611C……定価￥399,800
- CZ-611D……定価￥145,000
- CZ-6ST1……定価￥　5,800
- MD-2HD 20枚サービス
- 市販ゲームソフト2本サービス

他店には負けません TEL下さい 合計価格￥550,600

現金大特価!! (安いぞ)

安すぎて表示できません!!

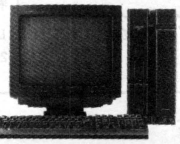


ACE セット
- CZ-601C……定価￥319,800
- CZ-601D……定価￥119,800
- CZ-6ST1……定価￥　5,800
- MD-2HD 20枚サービス
- 市販ゲームソフト2本サービス

OAランドで買わなきゃ損をする！ 合計価格￥445,400

現金大特価!! (推選)

安すぎて表示できません!!



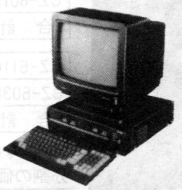
NEW X-1ターボZⅢセット
CRTクリーナー
キーボードカバープレゼント

Ⓐセット
- CZ-888CBK…定価￥169,800
- CZ-880DBK…定価￥ 99,800
- CZ-6ST1-B…定価￥　5,800 (チルトスタンド)
- MD-2HD 20枚サービス

合計定価￥275,400

現金価格 **特価中TEL下さい**

安すぎてゴメンなさい！



Ⓑセット
- CZ-888CBK…定価￥169,800
- CZ-830DBK…定価￥ 98,000
- CZ-6ST-1B…定価￥　5,800 (チルトスタンド)
- MD-2HD 20枚サービス

合計価格￥273,600

合計価格 **特価中TEL下さい**

安い!! X-1ターボⅡセット 単体販売OK

Ⓐセット
- CZ-881CBK…定価￥179,800
- CZ-880DBK…定価￥109,800
- CZ-6ST1-B…定価￥ 5, 800 (チルトスタンド)
- MD-2HD 20枚サービス

合計価格￥295,400

現金価格

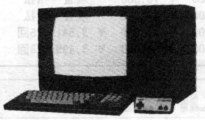


Ⓑセット
- CZ-881CBK…定価￥179,800
- CZ-830BK…定価￥ 98,000
- CZ-6ST1-B…定価￥　5,800 (チルトスタンド)
- MD-2HD 20枚サービス

合計価格￥283,600

現金価格

一度TEL下さい X-1Gセット 単体販売OK



X-1Gセット (お買得)
- CZ-822CB……定価￥118,000
- CZ-820DB……定価￥ 79,800
- MD-2D 20枚サービス

合計価格￥197,800

① X-1Gセット
＋パソコンラック
現金特価**￥81,000**

② X-1Gセット
＋市販ゲームソフト2本
現金特価**￥79,800**

周辺機器コーナー

X1用
- CZ-8BV2…定価￥ 39,800▶特価￥ **31,000**
- CZ-8BR1…定価￥ 29,800▶特価￥ **23,000**
- CZ-8DT2…定価￥ 44,800▶特価￥ **35,000**
- CZ-8BS1…定価￥ 23,800▶TEL下さい
- CZ-8TM2…定価￥ 49,800▶特価￥ **38,000**
- CZ-8EB3…定価￥ 33,800▶特価￥ **27,000**

X68000用
- CZ-6PU1A…定価￥ 38,000▶特価￥ **30,000**
- CZ-6BM1…定価￥ 26,800▶特価￥ **21,000**
- CZ-6BE1…定価￥ 88,000▶特価￥ **69,800**
- CZ-6VT1…定価￥ 69,800▶TEL下さい
- CZ-8NS1…定価￥188,000▶特価￥**149,000**
- CZ-6BC1…定価￥ 79,800▶特価￥ **63,000**

プリンターセットコーナー
①CZ-6PU1(カラービデオプリンター)定価￥198,000▶特価**￥152,000**
②CZ-8PC3(カラープリンター)……定価￥ 65,800▶特価￥ **53,000**
③CZ-8PK8(ドットプリンター)……定価￥152,000▶特価**￥115,000**
④CZ-8PK7(ドットプリンター)……定価￥122,000▶特価￥ **93,000**
⑤PC-PR201TH(カラープリンター)・定価￥145,000▶特価**￥103,000**
⑥PC-PR201G(ドットプリンター)……定価￥158,000▶特価￥ **99,000**

その他、周返機器・プリンター ソフトウェアー 20%～25% OFF!!

X68000用ソフトウェアー・コーナー
①CZ-212BS(BUSINESS)……定価￥ 68,000▶特価￥ **53,000**
②CZ-220BS(DATA)……定価￥ 58,000▶特価￥ **45,000**
③CZ-215MS(Sampling)……定価￥ 17,800▶特価￥ **13,800**
④CZ-221HS(NEW Print Shop)‥定価￥ 10,800▶特価￥ **15.500**
⑤CZ-227BS(TOP財務会計)……定価￥200,000▶特価**￥158,000**
⑥CZ-226BS(CARD)……定価￥229,800▶特価￥ **23,000**
⑦CZ-223CS(Communication)……定価￥ 19,800▶特価**￥115,500**
⑧CZ-213MS(MUSIC)……定価￥ 18,800▶特価￥ **14,800**
⑨CZ-211LS(C compiler)……定価￥ 39,800▶特価￥ **31,000**
⑩C-TRACE(キャスト)……定価￥ 68,000▶特価￥ **52,000**
⑪EW(イースト)……定価￥ 38,000▶特価￥ **29,000**

■ハードディスク ■特価品もありますのでTEL下さい。
- アイテック IT-MJ4(I/F付)……特価**￥98,000**
- アイテック IT-MJ4 C(I/F付)……特価**￥109,000**
- ウィンテック HD-404HS(I/F付)…特価**￥108,000**
- コンピュータ CRC-HD4A(I/F付)…特価**￥89,000**
- スナイパー SP-340(I/F付)……特価**￥92,000**
- アイテック ITH-320S(I/F付)……特価**￥79,800**
- ウィンテック HD-202(I/F付)……特価**￥58,000**
- スナイパー SR-520(I/F付)……特価**￥55,000**
- コンピュータ CRC-HD2A(I/F付)…特価**￥62,000**
- ロジテック LHD-32NR(I/F付)……特価**￥80,000**

今月の特価品 各一台限り
■A紙品(美品・POP品) ■B級品(キズ少々) ■C級品(キズ有り)

		A級品	B級品	C級品
X68000シリーズ	CZ-611C	￥282,000より	￥279,000	￥269,000
	CZ-601C	￥215,000より	￥210,000	￥205,000
	CZ-611D	￥104,000より	￥ 99,000	￥ 55,000
X-1ターボシリーズ	CZ-881	￥ 88,000より	￥ 85,000	￥133,000
MZシリーズ	MZ-2861	￥198,000より	￥185,000	￥158,000
その他	CZ-8NS1	￥145,000より	￥135,000	￥128,000

その他、いろいろありますのでTEL下さい!!

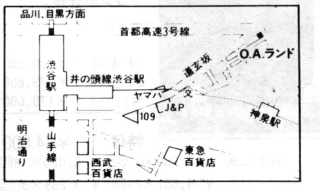
新年ディスケットセール
■D社クリーニングディスク付
- MD-2D 10枚 ￥ **850**
- MD-2DD 10枚 **￥1,150**
- MD-2HD 10枚 **￥1,180**
- MF-2D 10枚 **￥2,600**
- MF-2DD 10枚 **￥2,700**
- MF-2HD 10枚 **￥4,200**

中古パソコン(価格・在庫は変動します。予約は5日以内といたします。)

PC-9801VX2ト……**￥235,000**より
PC-9801VX2……**￥210,000**より
PC-9801VM2……**￥170,000**より
PC-980!VF2……**￥118,000**より
PC-9801M2……**￥145,000**より
PC-9801F2……￥ **88,000**より
PC-9801UV21……**￥148,000**より
PC-98LTM1(640KB)…￥ **89,000**より
PC-286,モデル0……**￥168,000**より
PC-286V-STD……**￥202,000**より
X-68000……**￥188,000**より
PC-8801mkII30……￥ **35,000**より
PC-8801mkIISR……￥ **73,000**より
PC-8801mkIIFR30…￥ **68,000**より
PC-8801mkIIMR……￥ **88,000**より
PC-88VA……**￥148,000**より
PC-8801mkIIFH30…￥ **85,000**より
PC-8801FA……**￥108,000**より
X-1Gモデル30……￥ **25,000**より
X-1ターボII……￥ **68,000**より
FM-77D2……￥ **28,000**より
FM-77AV2……￥ **42,000**より
FM-77AV20……￥ **52,000**より

通信販売のご案内
全国通販

■銀行振込で申し込みの方は商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。
〔振込先〕第一勧業銀行 渋谷支店
普通No.1163457 ㈱オーエーランド



■現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1～60回払で月々5,000円よりご自由に設定できます。

- 下取・買取は電話で見積りしております。責任を持って下取りさせて頂きます。
- ご注文、お問合せは…毎日午前10時から午後7時まで
- 商品のお届けは…入金確認後、即日発送致します。

㈱オーエー・ランド
〒150東京都渋谷区円山町20-4 第5日新ビル1F
☎(03)770-8855

関東エリアの送料は、1個につき￥1,000です。
※商品申し込み時、右の部分をお送り下さい。粗品を送ります。

1月 OAランド

パソコンプラザ OCT オクト

オクトで始まるパソコンワールド

☎ **03-730-6271**

●営業時間 AM 11:00～9:00/日曜・祭日PM7:00● 定休日:第2・3火曜定休
祭日の場合は翌日になります

〒144 東京都大田区蒲田4-6-7 FAX 03-730-6273

案内図



店頭セール実施中

全国通販 電話一本で、ハイ即納

OCT-1 システム インフォメーション

- ▶全商品保証付(メーカー保証)
- ▶超低金利ハッピークレジット(1回～60回)頭金ナシOK!
- ▶ボーナス一括払いOK!ボーナス2回払いOK!!
- ▶配達日の指定OK!(万全なサポート体制)
- ▶商品の組合せ自由! オクトフリーダムシステム
- ▶店頭デモンストレーション実施中

オクト
セレクテッドシステム

広告掲載商品以外の製品も取扱っております。

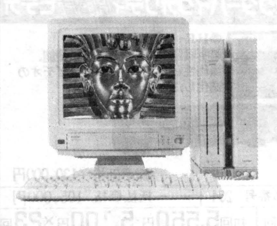
OCT-1 蒲田 HAPPY NEW YEARセール X68000フェア開催中!!

OPEN

《プレゼント実施中》★ドラゴンスピリッツ(¥8,800)★X-68000ポーチ ★3Mブラックディスク(MD-2HD10枚)

お好みのセットをお選び下さい。
送料無料

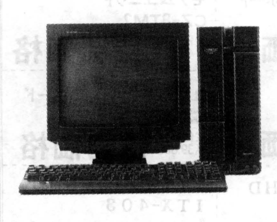
20MBハードディスク・モデル



X68000 ACE-HD

CZ-611C-GY/BK
定価¥399,800
現金特価!! お電話下さい。
推選

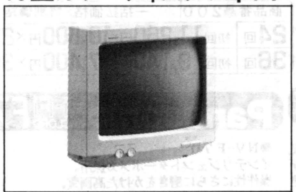
ハイコストパフォーマンス
FDモデル



X68000 ACE

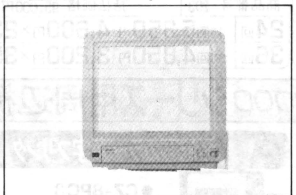
CZ-601C-GY/BK
定価¥319,800
現金特価!! お電話下さい。
推選

15型カラーディスプレイTV



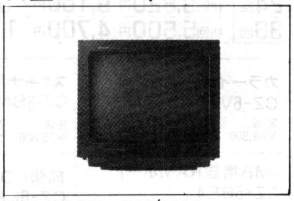
CZ-611D-GY/BK
定価¥145,000

15型カラーディスプレイTV



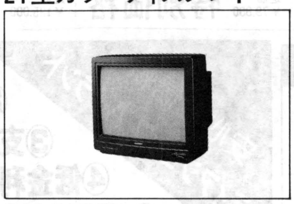
CZ-601D-GY/BK
定価¥119,800

14型カラーディスプレー



CZ-603D-GY/BK
定価¥84,800

21型カラーディスプレイ



CU-21CD
定価¥139,800

ACE-HD
①CZ-611C＋CZ-611D
合計金額¥544,800……………大特価TEL下さい。

全商品保証付(メーカー保証)

ACE
②CZ-601C＋CZ611D
合計金額¥464,800……………大特価TEL下さい。

超低金利ハッピークレジット(1回～60回)頭金なしOK!!

ACE-HD
③CZ-611C＋CZ-601D
合計金額¥519,600……………大特価TEL下さい。

ボーナス一括払いOK!ボーナス2回払いOK!

ACE
④CZ-601C＋CZ-601D
合計金額¥439,600……………大特価TEL下さい。

配達日の指定OK!(万全なサポート体制)

ACE-HD
⑤CZ-611C＋CZ-603D
合計金額¥484,600……………大特価TEL下さい。

商品の組合せ自由!オクトフリーダムシステム

ACE
⑥CZ-601C＋CZ-603D
合計金額¥404,600……………大特価TEL下さい。

店頭デモンストレーション実施中!

ACE-HD
⑦CZ-611C＋CU-21CD
合計金額¥539,600……………超特価TEL下さい。

オクト特選セット…電話は最後に

ACE
⑧CZ-601C＋CU-21CD
合計金額¥459,600……………超特価TEL下さい。

オクト推奨セット…電話はお早目に

※クレジットの回数は1回～60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット:送料¥2,000 ●店頭デモ実施中…専門の係員が詳細にアドバイス致します。ぜひご来店下さい。

オクトで始まるパソコンワールド。ニューイヤーセール開催中!! ご来店で、お買い上げの方には粗品プレゼント

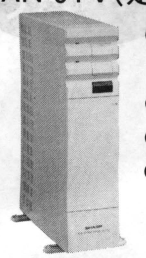
■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の
製品も取扱っております。

X68000周辺機器大セール実施中　全商品送料無料

パソコンチューナー
AN-8TV(定価35,800)

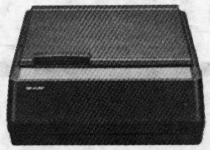


●CU-21CDでTVが見れる!!
●ビデオ入力端子
●モニター出力端子
●スーパーインポーズ表示可能

大特価 Cu-21CDとの組合せでドーゾ!!

カラーイメージスキャナ
CZ-8NS1(定価¥188,000)

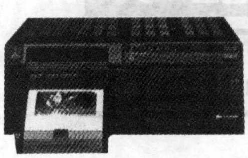
●A-4サイズまでの写真・図形フルカラー読み取り　●縮少・拡大自在



大特価 興味のある方TEL下さい

カラービデオプリンタ
CZ-6PV1(定価¥198,000)

●オリジナルCGや取り込んだ画像を色鮮やかにコピー!!



大特価 ビデオと組合せると最高!!

カラーイメージユニット
CZ-6VT1-BK(定価¥69,800)

●イメージ豊かなアートワークをサポート!! おしゃれなアートが楽しめます。

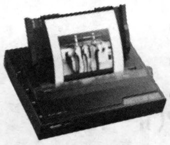


大特価 アート豊かな感性の方へ。

型名	商品	定価	特価
CZ-6BELA	IBM増設RAMボード	¥38,000	大特価
CZ-6BUI	ユニバーサルI/Oボード	¥39,800	大特価
CZ-6BGI	GP-IBボード	¥59,800	大特価
CZ-6BPI	プロセッサ・ボード	¥79,800	大特価
CZ-6BCI	FAXボード	¥79,800	大特価
CZ-6BMI	MIDボード	¥26,800	大特価

型名	商品	定価	特価
CZ-6EBI	拡張I/Oボックス	¥88,000	大特価
CZ-8TMZ	モデムユニット	¥49,800	大特価
CZ-6BN-I	スキャナ用パラレルボード	¥29,800	大特価
CZ-8NTI	トラックボール	¥13,800	大特価
AN-160SP	アンプ内蔵スピーカー	¥59,800	大特価

熱転写カラー漢字プリント　プリンター
CZ-8PC3 ¥65,800



●24ドットサーマルヘッド
●B5～B4縦まで
●ハガキ可能
●カラーハードコピー可能

大特価 オクト推選 TEL下さい!

①CZ-8PK7(24ピン80桁)
定価¥122,000
大特価・TEL下さい。

②CZ-8PK8 (24ピン136桁)
定価¥152,000
大特価・TEL下さい。

③CZ-8PK9
定価¥89,800
大特価・TEL下さい。

パソコンラック(4段)

推奨



キミだけのパソコン・スペースを作っちゃおう!!
移動自由自在
サイズ
1245(H)×614(D)×600(W) m/m
定価¥22,800
大特価¥12,000

X68000ソフト大セール実施中　※ゲームソフトオール23%off

〈グラフィック〉
●Z's STAFF PRO68K Ver.2.0
(シャフト)定価¥58,000
オクト特価**¥41,000**

〈データベース〉
●KAMIKAZE
(サムシンググッド)定価¥68,000
オクト特価**¥47,000**

〈グラフィック〉
●C-TRACE68
(キャスト)定価¥68,000
オクト特価**¥51,000**

型名	商品	定価	特価
BUSINESS PRO68K	統合型表計算	¥68,000	¥55,000
CARD PRO68K	カード型データベース	¥29,800	¥23,800
DATA	コマンド型データベース	¥58,000	¥45,000
COMMUNICATION PRO68K	通信ソフト	¥19,800	¥15,000
SAMPLING PRO68K	サンプリングエディタ	¥17,800	¥14,000
MUSIC PRO68K	楽譜ワープロ	¥18,800	¥15,000
SOUND PRO68K	サウンドエディタ	¥15,800	¥13,000
NEW PRINT SHOP PRO68K	ポップアートツール	¥19,800	¥15,000
C-COMPILER PRO68K	Cコンパイラ	¥39,800	¥32,000
EW	ワープロ	¥38,000	¥29,800
G-68	グラフィックツール	¥14,800	¥12,000
E-68K	スプライトエディタ	¥19,800	¥16,000

店頭ゲームソフトオール23%off! ビジネスソフト20%より特価中

★通信販売お申込みのご案内★　〒144 東京都大田区蒲田4-6-7　TEL:03-730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。●入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込:お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留:封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

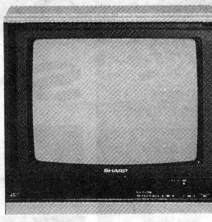
振込先
三菱銀行 蒲田支店 当 No.0278691
東京都民銀行 蒲田支店 当 No.0320955
株式会社 億人(オクト)

▶ソフトのご注文は、OCT-2にて取扱っておりますので、当誌上380Pをご覧下さい。
※掲載の価格は11/20現在ですので、まずは、お電話にてご確認ください。

オクトで始まるパソコンワールド。ニューイヤーセール開催中!! ご来店で、お買い上げの方には粗品プレゼント



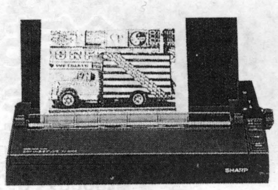
SHARP
CZ-880DGY 新品同様
(14インチ400/200RGBTV)
¥109,800➡ **¥69,800**
(色はグレーになります。)



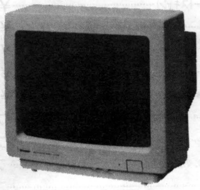
SHARP
CZ-820DE・B 新品
(14インチ2000字RGBTV)
¥79,800➡ **¥39,800**



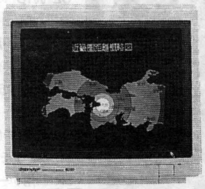
SHARP
CZ-880CB (X-1TurboZ本体)
¥218,000➡ **¥74,800**
CZ-880DB 新品同様
¥109,800➡ **¥85,000**
セット価格
¥327,800➡ **¥159,800**



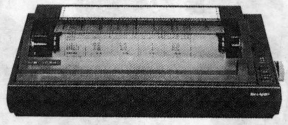
SHARP
CZ-8PC2 新品同様
(10インチ熱転写カラー漢字プリンタ)
¥69,800➡ **¥44,800**



SHARP
CU-14A4 新品
(14インチ4050字アナログ・デジタルRGB、PC用アナログRGBケーブル付)
¥89,800➡ **¥49,800**



SHARP
CU-14GB/E 新品
(14インチ2000字デジタルRGB)
¥49,800➡ **¥29,800**



SHARP
CZ-8PK6 新品同様
(15インチ漢字プリンタ)
¥159,000➡ **¥59,800**



SHARP
CZ-822C
(X-1Gモデル30本体) 新品同様
¥118,000➡ **¥49,800**
X-1Gモデル30TVディスプレイセット
(本体+CZ-820D) 特選極上品
¥197,800➡ **¥79,800**

SHARP

本体

CZ-812C(X-1F model 20)	¥139,800➡	**¥32,000**
CZ-822C(X-1G model 30)	¥118,000➡	**¥42,000**
CZ-830C(X-1Twin)	¥99,800➡	**¥58,000**
CZ-850CB(X-1Turbo モデル10)	¥168,000➡	**¥25,000**
CZ-870CB(X-1TurboⅢ)	¥168,000➡	**¥55,000**
CZ-880CB(X-1Turbo Z)	¥218,000➡	**¥74,800**

ディスプレイ

12M-314C(14"カラー4050文字)	¥128,000➡	**¥45,000**
14M-111C(14"カラー1000文字)	¥67,800➡	**¥15,000**
14M-142C(14"カラー2000文字)	¥99,800➡	**¥22,000**
14M-511C(14"カラー2000文字)	¥59,800➡	**¥20,000**
CZ-880DGY(14"カラー2000文字RGBTV)	¥109,800➡	**¥64,800**
CU-14BD(14"カラー4050文字)	¥64,800➡	**¥40,000**

ディスクドライブ・プリンタ・他

CZ-503F(5"2D、1ドライブ)	¥49,800➡	**¥25,000**
CZ-81P(ミニサイズプリンタ)	¥34,800➡	**¥10,000**
CZ-8PP2(カラープロッタプリンタ) 新品	¥54,800➡	**¥15,000**
CZ-8PD2(10"ドットプリンタ)	¥79,800➡	**¥28,000**
CZ-8PD3(10"ドットプリンタ)	¥59,800➡	**¥28,000**
CZ-8PC1(10"24ドットカラー漢字熱転写プリンタ)	¥69,800➡	**¥35,000**
CZ-8PC2(10"24ドットカラー漢字熱転写プリンタ)	¥69,800➡	**¥38,000**
CZ-8PC2(10"24ドットカラー漢字熱転写プリンタ) 新品	¥69,800➡	**¥44,800**
CZ-8PK6(15"24ドット漢字プリンタ) 新品	¥159,000➡	**¥59,800**
MZ-1P06(80桁漢字プリンタ)	¥234,000➡	**¥45,000**
MZ-1P09(MZ-1500用カラープロッタプリンタ)	¥39,800➡	**¥10,000**
MZ-1P17(80桁24ドットカラー漢字熱転写プリンタ・CZ用ケーブル付) 新品	¥76,600➡	**¥42,800**
MZ-1P17(80桁24ドットカラー漢字熱転写プリンタ・MZ用ケーブル付) 新品	¥76,600➡	**¥46,800**
CZ-8SS2(システムスタンド)	¥5,500➡	**¥4,000**
CZ-8BS1(FM音源ボード) 新品	¥23,800➡	**¥20,000**

＊SHARP X-1シリーズ特選極上品コーナー＊

CZ-820CE(X-1G/10) 新品同様	¥69,800➡	**¥9,800**
CZ-822CB(X-1G/30) 新品同様	¥118,000➡	**¥49,800**
CZ-822C+CZ-822D(X-1G/30セット) 特選極上品	¥197,800➡	**¥79,800**

＊SHARP ディスプレイ特選極上品コーナー＊

CU-14G(14"カラー2000文字) 新品	¥49,800➡	**¥29,800**
CU-14ED(14"カラー4050文字) 新品	¥79,800➡	**¥49,800**
CZ-820D(14"カラー2000文字RGBTV) 新品同様	¥79,800➡	**¥39,800**
CZ-880DB(15"カラー4050文字RGBTV) 新品同様	¥109,800➡	**¥85,000**
CZ-880DGY(15"カラー4050文字RGBTV) 新品同様	¥109,800➡	**¥69,800**
CZ-600D(15"カラー4050文字RGBTV) 新品同様	¥129,800➡	**¥88,000**

6つの安心のアフターサービス

1 C.B.クラブ

■あなたも今すぐ会員に!!

当社で商品をお買い上げの方全員に、C.B.クラブカードを無料でお送り致します。このカードをお持ちの方ならば次の買い換え時や、付属品の購入時に会員特別価格でご購入になれます。

2 C.B.サポートホットライン

☎03(797)1234

■トラブルへの対応!!

当社でコンピュータをお買い上げいただいたお客様に万一、トラブルが発生した場合、このホットラインで親切に対応いたします。

3 C.B.レスキューシステム

■迅速なサポート体制!!

お客様のお手元でトラブルが発生した場合、当社より引取りにお伺い致します。万一、お買いになった機械が故障しても安心です。

4 C.B.クイック・チェンジシステム

■新品交換体制も万全!!

お買い上げになったパソコンが、万一初期不良でも安心です。商品到着後7日以内にご連絡いただければ、新品と交換致します。

5 RX2アフターサポート

■PC-9801愛好家にお得です!!

NEC RX2をお買い上げいただいたお客様に保証期間中、万一故障があった場合無料で代品を貸出します。

6 C.B.Q&Aホットライン

☎03(797)1233

■素朴な疑問何でもどうぞ!!

ハードウェア、ソフトウェアに関するご質問なら内容を問わずどなたからでも親切に、ご相談をお受け致しております。

●電話一本で高額下取り、即商品はお手元へ!
●あなたの不要になったパソコンを電話一本で査定し買取ります。
●掲載の商品以外も取り扱っておりますのでお気軽にお電話下さい。

▼本社注文デスク
03(797)1221

コンピュータバンク

株式会社パシフィックコンピュータバンク 〒150 東京都渋谷区渋谷1-6-8 井上ビル 営業時間/AM9:30～PM9:30 年中無休

Let's CALL
超優良中古パソコンが電話一本で買える!!
03(797)1221

今年もさらに充実したサービスで安心をお届けいたします。

ありがとう70000人！
70000人もの方々が体感したIPL安心の保証



山下章先生講演風景
at IPLショールーム

70,000人もの人々が体感した安心感。
――信頼のIPLワイドサポート

●業界初、IPLでこそ成し得た3倍保証。
メーカー保証12ヶ月の商品なら36ヶ月の保証と長期間の保証を実施。末長く安心してご利用いただけるよう、IPLが成し得たワイドなサポート体制。

●IPLだからこそ初期不良への保証も万全。交換期間も1ヶ月ともっとも長期間です。IPLだからこそ安心が長続きします。

こんなにかかる修理費用
プリンタヘッド交換￥29,500以上/98シリーズメインボード交換￥21,600以上/ドライブ交換￥13,200以上

比べてほしいから、ご紹介します。
―さらにお買得IPLクレジット

●ステップアップクレジットがおトク。
まず月々1,000円からスタートして2年後から3,000円アップ。ボーナスも1年後1万円。3年後3万円。また夏のボーナスを貯金して冬のボーナスからのお支払いも大丈夫。夏・冬のボーナスどちらか一つをセレクト。ボーナス年一回だけもOK。
システムはすぐお手元へ。冬のボーナス一括、冬夏ボーナス2括払いもOK。

●追加購入もクレジットだから便利。
追加購入も買い換えもご利用中のIPLクレジットを月々僅か1,000円ずつの調整でOK。

●IPLは8月21日ついに70,001人めのお客様を迎えました。

ORDER TELEPHONE
電話受付：AM10:00～PM8:00 水曜定休日

本　社　0467-24-7511
大　阪　06-311-2736

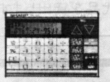
銀　座　03-541-3058　青　山　03-470-0061　札　幌　011-621-1444
仙　台　022-266-0531　広　島　082-293-7881　福　岡　092-481-2644

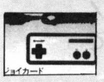
商品管理部（納期、配達日のお問合せ、ご指定日のご連絡）　0467-24-1154
メンテナンス部（ハード上のご相談、お問合せ、初期不良の対応）　0467-24-0453
FAX（ご注文、お見積り、カタログ編集などスピーディに）　0467-24-0561
タイムリーボックス（ホットな新製品ニュースをお知らせします。）　0467-24-0941

ご注文お問合せ
――0467-24-1154
下取りホットライン
――0467-24-2040

6のチャンスで魅力倍増！
今お買い上げの方に、うれしいプレゼント進呈中!!

1　期間中、システムでお買い上げの方、先着200名様に、電話帳電卓をプレゼント。（電話番号・スケジュールを記憶、10桁電卓機能付）
2　期間中、デスク（SA-600）をお買上げの方全員に、A-300（**原稿用スタンド**￥8,000）をプレゼント。
3　期間中、シャープ製品をシステムでお買上げの方にCZ-8NJ1（**ジョイカード**）をプレゼント。
4　期間中、エプソンAP800お買い上げの方全員にリボンパック（金2本銀1本）をプレゼント
5　期間中、エプソンAP-800（シャープ用）をお買上げの方全員に**ケーブル**（￥8,800）得々パックをプレゼント
6　期間中、セットでお買上げのご希望の方に**モデム**（SR-30￥19,800）をプレゼント




ジョイカード



SHARP
SHARP X-68000

アクセス No.X0202
価格￥489,000 ➡ IPL超特価

CZ-601C（CPU68000,2Mバイト,65536同時発色）	￥319,800
CZ-601D（.39ミリ、アナログ3モードオートスキャン）	￥119,800
源平討魔伝	￥ 7,800
ドラゴンスピリット	￥ 8,800
CZ-218AS（沙羅曼蛇）	￥ 8,800
3M ブランクディスケット（5"2HD＊10枚）	￥ 24,000
CZ-8NJ1（ジョイカード）	プレゼント
CZ-6ST1（角度自由自在、調節OK!）	プレゼント
パーソナルモデムSR-30（パソコンの新しい可能性をプレゼント!￥19800）	プレゼント
SR-30用ケーブル（￥5000）	プレゼント
X68通信講座（信頼の"サポート"テスト問題付。ひとりひとりをしっかりフォロー）	￥ 0
初期不良期間（ワイドに1ケ月間の交換システム!）	￥ 0
安心の3倍保証（IPL保証書付き）	￥ 0

標準価格￥489,000
￥2,700×72回　ボーナス　2.0万×12回

￥ 3,600×60回	ボーナス	2.0万×10回
￥ 3,300×48回	ボーナス	3.0万×8回
￥ 5,700×36回	ボーナス	3.0万×6回
￥ 6,800×24回	ボーナス	3.5万×4回

アクセス No.X0205
価格￥493,400 ➡ IPL超特価

CZ-601C（CPU68000,2Mバイト,65536同時発色）	￥319,800
CZ-603D（超高解像度0.31ドット14"チルト台付）	￥ 84,800
源平討魔伝	￥ 7,800
信長の野望/全国版	￥ 9,800
A列車で行こうII	￥ 12,800
ラストハルマゲドン	￥ 9,800
名監督II	￥ 9,800
三国志	￥ 14,800
3M ブランクディスケット（5"2HD＊10枚）	￥ 24,000
CZ-8NJ1（ジョイカード）	プレゼント中
X68通信講座（信頼の"サポート"テスト問題付ひとりひとりをしっかりフォロー）	￥ 0
パーソナルモデムSR-30（パソコンの新しい可能性をプレゼント!￥19800）	プレゼント中
SR-30用ケーブル（￥5000）	プレゼント中
初期不良期間（ワイドに1ケ月間の交換システム!）	￥ 0
安心の3倍保証（IPL保証書付き）	￥ 0

標準価格￥493,400
￥2,900×72回　ボーナス　2.0万×12回

￥ 3,900×60回	ボーナス	2.0万×10回
￥ 3,700×48回	ボーナス	3.0万×8回
￥ 6,200×36回	ボーナス	3.0万×6回
￥10,000×24回	ボーナス	3.8万×4回

アクセス No.X0207
価格￥529,400 ➡ IPL超特価

CZ-601C（CPU68000,2Mバイト,65536同時発色）	￥319,800
CZ-601D（.39ミリ、アナログ3モードオートスキャン）	￥119,800
CZ-8PK9（24ピン80桁漢字プリンタ,ハガキ可,トラクタ付）	￥ 89,800
CZ-8NJ1（ジョイカード）	プレゼント中
CZ-6ST1（角度自由自在、調節OK!）	プレゼント中
パーソナルモデムSR-30（パソコンの新しい可能性をプレゼント!￥19800）	プレゼント中
SR-30用ケーブル（￥5000）	プレゼント中
X68通信講座（信頼の"サポート"テスト問題付。ひとりひとりをしっかりフォロー）	￥ 0
初期不良期間（ワイドに1ケ月間の交換システム!）	￥ 0
安心の3倍保証（IPL保証書付き）	￥ 0

標準価格￥529,400
￥3,000×72回　ボーナス　2.28万×12回

￥ 4,600×60回	ボーナス	2.0万×10回
￥ 4,500×48回	ボーナス	3.0万×8回
￥ 7,300×36回	ボーナス	3.0万×6回
￥ 9,500×24回	ボーナス	5.0万×4回

初期不良交換期間1ヶ月

超低金利……　組合せ自由　全国無料配送

※今回掲載の製品は、1月18日より2月18日までの期間に限らせていただきます。

アクセス No.X0201

価格¥546,200 ➡ IPL超特価

CZ-601C(CPU68000, 2Mバイト, 65536同時発色)	¥319,800
CZ-603D(超高解像度0.31ドット14"チルト台付)	¥ 84,800
CZ-8PC3(10"カラー熱転写, ハガキ可。漢字53字/秒)	¥ 65,800
CZ-218AS(沙羅曼蛇)	¥ 8,800
ドラゴンスピリット	¥ 8,800
ラストハルマゲドン	¥ 9,800
こはく色の遺言	¥ 9,800
スペース ハリアー	¥ 6,800
源平討魔伝	¥ 7,800
3M ブランクディスケット(5"2HD*10枚)	¥ 24,000
CZ-8NJ1(ジョイカード)	プレゼント
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント!¥19800)	プレゼント
SR-30用ケーブル(¥5000)	プレゼント
X68通信講座(信頼の"サポート"テスト問題付。ひとりひとりをしっかりフォロー)	¥ 0
初期不良期間(ワイドに1ケ月間の交換システム!)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

標準価格¥546,200

¥2,800 ×72回 ボーナス 2.5万×12回

¥ 3,100×60回	ボーナス 3.0万×10回
¥ 4,700×48回	ボーナス 3.0万×8回
¥ 7,600×36回	ボーナス 3.0万×6回
¥ 9,900×24回	ボーナス 5.0万×4回

アクセス No.X0206

価格¥510,000 ➡ IPL超特価

CZ-601C(CPU68000, 2Mバイト, 65536同時発色)	¥319,800
CU-21CD(迫力の21"カラーアナログCRT3モードマルチスキャン方式)	¥139,800
源平討魔伝	¥ 7,800
ドラゴンスピリット	¥ 8,800
ラストハルマゲドン	¥ 9,800
3M ブランクディスケット(5"2HD*10枚)	¥ 24,000
CZ-8NJ1(ジョイカード)	プレゼント中
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント!¥19800)	プレゼント中
SR-30用ケーブル(¥5000)	プレゼント中
X68通信講座(信頼の"サポート"テスト問題付。ひとりひとりをしっかりフォロー)	¥ 0
初期不良期間(ワイドに1ケ月間の交換システム!)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

標準価格¥510,000

¥2,900 ×72回 ボーナス 2.0万×12回

¥ 3,800×60回	ボーナス 2.0万×10回
¥ 3,600×48回	ボーナス 3.0万×8回
¥ 5,000×36回	ボーナス 3.65万×6回
¥ 9,800×24回	ボーナス 3.8万×4回

SHARP X68000 ACE HD

アクセス No.X0209

価格¥673,600 ➡ IPL超特価

CZ-611C(20MHDD搭載、65536色発色、FM8音源内蔵)	¥399,800
CZ-603D(超高解像度14"チルト台付)	¥ 84,800
CZ-6BE1A(1MB増設RAMボード)	¥ 38,000
CZ-213MS(MUSIC PRO 68K)	¥ 18,800
CZ-214MS(SOUND PRO 68K)	¥ 15,800
CZ-215MS(ADPCM機能をサポートした、サンプリングエディタ)	¥ 17,800
CZ-8PC3(10"カラー熱転写, ハガキ可。漢字53字/秒)	¥ 65,800
CZ218AS(沙羅曼蛇)	¥ 8,800
3Mブランクディスケット(5"2HD*10枚)	¥ 24,000
電話帳電卓*贈呈(電話番号50人分, スケジュルメモOK 電卓機能付)	プレゼント中
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント!¥19800)	プレゼント中
SR-30用シャープケーブル(¥5000)	プレゼント中
X68通信講座(信頼の"サポート"テスト問題付。ひとりひとりをしっかりフォロー)	¥ 0
初期不良期間(ワイドに1ケ月間の交換システム!)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

標準価格¥673,600

¥3,000 ×72回 ボーナス 3.43万×12回

¥ 5,000×72回	ボーナス 2.23万×12回
¥ 5,100×60回	ボーナス 3.0万×10回
¥ 7,200×48回	ボーナス 3.0万×8回
¥ 9,800×36回	ボーナス 3.5万×6回

アクセス No.X0204

価格¥671,600 ➡ IPL超特価

CZ-611C(20MHDD搭載、65536色発色、FM8音源内蔵)	¥399,800
CZ-611D(.31ミリ、アナログ3モードオートスキャン)	¥145,000
CZ-213MS(MUSIC PRO 68K)	¥ 18,800
CZ-214MS(SOUND PRO 68K)	¥ 15,800
CZ-8PC3(10"カラー熱転写, ハガキ可。漢字53字/秒)	¥ 65,800
ザ.リターン.オブ.イシター(エス.ビー.エス)	¥ 7,800
ドラゴンスピリット(マイコンソフト)	¥ 8,800
信長の野望全国版(光栄 5"2HD)	¥ 9,800
CZ-8NJ1(ジョイカード)	プレゼント中
CZ-6ST1(上下左右角度調整自在チルト台¥5800)	プレゼント中
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント!¥19800)	プレゼント中
SR-30用ケーブル(¥5000)	プレゼント中
X68通信講座(信頼の"サポート"テスト問題付。ひとりひとりをしっかりフォロー)	¥ 0
初期不良期間(ワイドに1ケ月間の交換システム!)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

標準価格¥671,600

¥3,000 ×72回 ボーナス 3.55万×12回

¥ 4,800×72回	ボーナス 2.5万×12回
¥ 5,000×60回	ボーナス 3.25万×10回
¥ 7,500×48回	ボーナス 3.0万×8回
¥10,000×36回	ボーナス 3.65万×6回

アクセス No.X0208

価格¥972,470 ➡ IPL超特価

CZ-611C(20MHDD搭載、65536色発色、FM8音源内蔵)	¥399,800
CZ-611D(.31ミリ、アナログ3モードオートスキャン)	¥145,000
3M ブランクディスケット(5"2HD*10枚)	¥ 24,000
CZ-6ST1(角度自由自在、調節OK!)	¥ 5,800
CZ-211LS(C compilerソフト開発を効率良くサポート)	¥ 39,800
CZ-8NS1(フルカラーA4ズーム機能色ずれの少ない線順次方式ソフト付き)	¥188,000
CZ-6BN1(68000用スキャナ用パラレルボード)	¥ 29,800
CZ-6BC1(I/Oスロットに装着パソコンからFAX送受信A4B4GIII)	¥ 79,800
CZ-8PC3(10"カラー熱転写, ハガキ可。漢字53字/秒)	¥ 65,800
A4カット紙(100枚)	¥ 470
電話帳電卓*贈呈(電話番号50人分, スケジュルメモOK 電卓機能付)	プレゼント
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント!¥19800)	プレゼント
SR-30用ケーブル(¥5000)	プレゼント
X68通信講座(信頼の"サポート"テスト問題付。ひとりひとりをしっかりフォロー)	¥ 0
初期不良期間(ワイドに1ケ月間の交換システム!)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

標準価格¥972,470

¥4,300 ×72回 ボーナス 5.0万×12回

¥ 7,600×72回	ボーナス 3.0万×12回
¥ 9,600×60回	ボーナス 3.0万×10回
¥ 9,200×48回	ボーナス 5.0万×8回
¥14,200×36回	ボーナス 5.0万×6回

アクセス No.X0203

価格¥608,800 ➡ IPL超特価

CZ-601C(CPU68000, 2Mバイト, 65536同時発色)	¥319,800
CZ-611D(.31ミリ、アナログ3モードオートスキャン)	¥145,000
CZ-221HS(NEW Prints Shop様々なカードなどを自由に作成)	¥ 19,800
CZ-215MS(AD PCM機能をサポートしたサンプリングエディタ)	¥ 17,800
CZ-8PC3(10"カラー熱転写, ハガキ可。漢字53字/秒)	¥ 65,800
CZ-232AS(熱血高校ドッジボール部)	¥ 7,800
CZ-218AS(沙羅曼蛇)	¥ 8,800
3M ブランクディスケット(5"2HD*10枚)	¥ 24,000
CZ-8NJ1(ジョイカード)	プレゼント
CZ-6ST1(上下左右角度調節自在チルト台¥5800)	プレゼント
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント!¥19800)	プレゼント
SR-30用ケーブル(¥5000)	プレゼント
X68通信講座(信頼の"サポート"テスト問題付。ひとりひとりをしっかりフォロー)	¥ 0
初期不良期間(ワイドに1ケ月間の交換システム!)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

標準価格¥608,800

¥2,700 ×72回 ボーナス 3.0万×12回

¥ 4,400×72回	ボーナス 2.0万×12回
¥ 4,800×60回	ボーナス 2.5万×10回
¥ 5,700×48回	ボーナス 3.0万×8回
¥ 8,000×36回	ボーナス 3.47万×6回

安心の3倍保証

COMPUTER CREATION
IPL
株式会社・アイピーエル
〒248 鎌倉市雪ノ下4−1−12
雪ノ下ビル
鎌倉市雪ノ下3-4-23：商品管理部
AM10:00▶PM8:00
水曜日定休



アクセス No.X0210

価格¥675,200 ➡ IPL超特価

CZ-611C(20MHDD搭載、65536色発色、FM8音源内蔵)	¥399,800
CZ-601D(.39ミリ、アナログ3モードオートスキャン)	¥119,800
CZ-221HS(NEW PrintShop様々なカードなどを自由に作成)	¥ 19,800
EW & E1(漢字変換フロントプロセッサ搭載, 高速日本語ワープロ)	¥ 38,000
AP-800(初!美しい印字48ドットカラー漢字熱転写はがきからB4縦可)	¥ 97,800
シャーププリンタケーブル	プレゼント中
APCRP2(金2本銀1本リボンパック, お祝い, 案内, 挨拶状, カード)	プレゼント中
CZ-6ST1(上下左右角度調節自在チルト台¥5800)	プレゼント中
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント!¥19800)	プレゼント中
SR-30用ケーブル(¥5000)	プレゼント中
X68通信講座(信頼の"サポート"テスト問題付。ひとりひとりをしっかりフォロー)	¥ 0
初期不良期間(ワイドに1ケ月間の交換システム!)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

標準価格¥675,200

¥3,100 ×72回 ボーナス 3.5万×12回

¥ 5,000×72回	ボーナス 2.31万×12回
¥ 7,000×60回	ボーナス 2.0万×10回
¥ 7,400×48回	ボーナス 3.0万×8回
¥ 9,800×36回	ボーナス 3.7万×6回

アクセス No.X0211

価格¥776,870 ➡ IPL超特価

CZ-611C(20MHDD搭載、65536色発色、FM8音源内蔵)	¥399,800
CZ-603D(超高解像度0.31ドット14"チルト台付)	¥ 84,800
CZ-6BE1A(1MB増設RAMボード/CZ-601, 611用)	¥ 38,000
CZ-8NS1(フルカラーA4ズーム機能色ずれの少ない線順次方式ソフト付き)	¥188,000
CZ-8PC3(10"カラー熱転写, ハガキ可。漢字53字/秒)	¥ 65,800
A4カット紙(100枚)	¥ 470
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント!¥19800)	プレゼント中
SR-30用ケーブル(¥5000)	プレゼント中
X68通信講座(信頼の"サポート"テスト問題付。ひとりひとりをしっかりフォロー)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

標準価格¥776,870

¥1,900 ×72回 ボーナス 5.0万×12回

¥ 5,000×72回	ボーナス 3.14万×12回
¥ 6,900×60回	ボーナス 3.0万×10回
¥ 8,000×48回	ボーナス 3.75万×8回
¥10,100×36回	ボーナス 5.0万×6回

ビギナーズ・ホットライン

初心者の方々のために、常設の無料相談窓口を設けました。お気軽にご利用ください。

0467−24−0941

日曜・祭日・指定日配送OK！　輸送上のトラブルにも対応

※IPL超特価システムはさらにおトクなシステム。お電話でお問合せください。

Jan.18～Feb.18

J&P オプション ソフト 通信販売

社団法人日本通信販売協会 正会員店

全国どこでも 無料配達

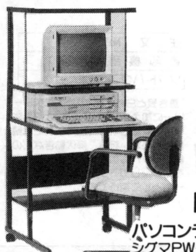


パソコン通信

J&P HOT LINE でもお申し込みいただけます。

J&Pメールショッ

■600%幅パソコンラック



X2-300

パソコンラック
シグマPW-833 ¥29,500
J&P特価 16,800円
W600×D640×H1280%



¥32,800を
J&P特価 15,800円

X2-301

パソコンラック
エレコムDS-20 ¥32,800
W650×D700×H1260%
●ロック式キャスター付
●コード落しボックス付



¥43,000を
J&P特価 19,800円

X2-302

パソコンラック
エレコムPD-02 ¥43,000
W640×D700×H1305%
●ロック式キャスター付●コード落しボックス付
●中棚が2枚でキーボード収納棚又はペーパー置きにできます。
●2Pコンセント付



X2-303

パソコンラック
エレコムPD-01 ¥45,000
J&P特価 39,800円
W650×D700×H1245%
●ロック式キャスター付●コード
●コード落しボックス付



X2-304

トール・ステーション・デス
シグマPW-6502 ¥44,000
J&P特価 39,800円
W650×D650×H1345%
●ロック式キャスター付●パ
ンチェアー、原稿台別売



X2-305

パソコンラック
エレコムDS-10 ¥32,000
J&P特価 16,800円
W650×D710×H1285%
●ロック式キャスター付
●コード落しボックス付
●コンセント付
●2Pコンセント2個付



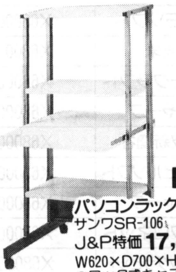
X2-306

パソコンラック
コニカED-50 ¥29,800
J&P特価 14,800円
W640×D700×H1260%
●ロック式キャスター付
●コードクランプ付



X2-307

パソコンラックER-600
オーバートップデスクER-606付
エレコムER-600
オプションプリンタ台ER-606
J&P特価 15,800円 ¥合計34,980
W650×D625×H1355%
●2Pコンセント2個付



X2-308

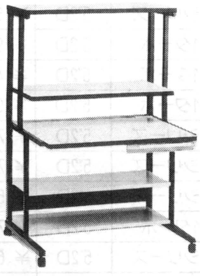
パソコンラック
サンワSR-106 ¥36,800
J&P特価 17,800円
W620×D700×H1265%
●ロック式キャスター付●コード落しボックス付●コンセント付
●キャスター付



X2

ラックI model 2
サンワRAC-312 ¥38,800
J&P特価 36,800円
W662×D675(キーボード収
580×H1186%
●キャスター付
●キーボード収納トレー付

■900%幅パソコンラック



パソコンデスク
X2-310
エレコムPD-99
J&P特価 29,800円 ¥48,000
W900×D700×H1280%
●ロック式キャスター付●コード落しボックス付●コンセント付●B4判引き出し別売¥3,500



¥49,800を
J&P特価 42,900円

X2-312

ワーク・ステーション・デスク
シグマPW-9300 ¥49,800
W900×D740×H875
●コンセント、手許スイッチ付●5"フロッピィサイズ引出し●データスタンド、ペーパーストレイジOAチェアー別売



パソコンデスク
X2-311
エレコムER-900
J&P特価 29,800円 ¥43,000
W900×D700×H1270%
●ロック式キャスター付●B4判引き出し付



パソコンデスク
X2-313
サンワSR-107
J&P特価 29,800円 ¥41,800
W940×D700×H1265%
●ロック式キャスター付●コード落し付
●2Pコンセント2個付

■1,200%幅パソコンラック



X2-314

パソコンデスク
エレコムPD-120 ¥48,000
J&P特価 31,500円
W1200×D700×H820~1180%
●ロック式キャスター付●オーバートップ調節可●コンセント付●B4判引き出し別売 ¥3,500



X2-315

¥48,000を
J&P特価 28,000円

パソコンデスク
エレコムER-1200 ¥48,000
W1200×D700×H820~1180%
●ロック式キャスター付
●オーバートップデスク高さ調節



98ステーションD-II
X2-316
サンワDSF-992L ¥59,800
J&P特価 56,800円
W1200×D800×H840~1190%
●手元集中スイッチコンセント付
●オーバートップ調節可●コード落し●ブックエンド付

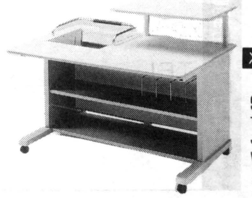


X2-317

パソコンデスク
エレコムEDX-1212 ¥69,800
J&P特価 62,800円
W1200×D800×H680(+460)%
●手許集中スイッチコンセント付
●ロック式キャスター付●中棚は上下5段階調節可●オーバートップデスク付

X2-318

トレーユニット(オプション)
エレコムFO-50E ¥4,600
J&P特価 4,200円
W377×D270×H37%
B4判サイズで、よく使う原稿やプリント用紙を収納するのに便利です。



X2-320

98ステーションD-V
サンワDSF-995L ¥69,800
J&P特価 66,500円
W1200×D800×H840~1190%
●手許集中スイッチコンセント付
●オーバートップ調節可
●コード落し●ブックエンド付

X2-319

ペーパーストレイジ(オプション)
エレコムPO-90E ¥13,000
J&P特価 12,300円
15×11インチ連続用紙がスムーズにIN-OUTします。

¥はメーカー標準価格です。



クリエイト特典

- ●全商品完全保証書付(メーカー保証)
- ●全国無料配達(一部離島の方は有料になります)
- ●配達日の指定OK(日曜・祭日にかかわらずお客様のご都合にあわせて配達します)
- ●どんな商品の組合せも自由自在(ご予算、用途に応じ自由自在にシステムアップできます)
- ●中古パソコン高額下取り(今お使いのパソコンをわずかな差額でグレードアップ)
- ●お支払い方法自由(低金利の均等払い、ボーナス一括払いもご利用ください)

営業時間(年中無休)
AM10:00〜PM7:00(日曜・祭日はPM6:00まで)

当社はX68000の販売認定店です。どんなことでも安心してご相談ください。

(★X68000をお買上げのお客様にもれなく、▶X68000オリジナルテレホンカードプレゼント!!)

X68000 ACE 基本セット

●CZ-601C(本体+キーボード)	¥319,800
●CZ-601D(カラー専用ディスプレイ)	¥119,800
●CZ-6ST1(チルトスタンド)	¥5,800
●ブランクディスケット(5"2HD・10枚)	¥10,000
●ソフト/アルカノイド	¥サービス
■定価合計 ¥455,400▶	クリエイト特価

電話にてお問合せください。

X68000 ACE 格安基本セット

●CZ-601C(本体+キーボード)	¥319,800
●CZ-603D(カラー専用ディスプレイ)	¥84,800
●CZ-8PC3(熱転写カラー漢字プリンタ)	¥65,800
●ソフト/アルカノイド	¥サービス
●プリンタ用紙	¥サービス
●ブランクディスケット(5"2HD・10枚)	¥10,000
■定価合計 ¥480,400▶	クリエイト特価

電話にてお問合せください。

X68000 ACE HD 基本セット

●CZ-611C(本体+キーボード)	¥399,800
●CZ-601D(カラー専用ディスプレイ)	¥119,800
●CZ-6ST1(チルトスタンド)	¥5,800
●ブランクディスケット(5"2HD・10枚)	¥10,000
●ソフト/アルカノイド	¥サービス
■定価合計 ¥535,400▶	クリエイト特価

電話にてお問合せください。

X68000 ACE Vーセット

●CZ-601C(本体+キーボード)	¥319,800
●CZ-601D(カラー専用ディスプレイ)	¥119,800
●CZ-8PC2(熱転写カラー漢字プリンタ)	¥65,800
●CZ-6TV1(カラーイメージユニット)	¥69,800
●CZ-6ST1(チルトスタンド)	¥サービス
●ブランクディスケット(5"2HD・10枚)	¥10,000
●ソフト/アルカノイド	¥サービス
■定価合計 ¥585,200▶	クリエイト特価

電話にてお問合せください。

X68000 ACE ミュージックワークセット

●CZ-601C(本体+キーボード)	¥319,800
●CZ-601D(カラー専用ディスプレイ)	¥119,800
●CZ-8PC3(熱転写カラー漢字プリンタ)	¥65,800
●SOUND PRO-68K(音色作成ツール)	¥15,800
●CZ-6ST1(チルトスタンド)	¥5,800
●ブランクディスケット(5"2HD・10枚)	¥10,000
●MUSIC PRO-68(楽譜入力ツール)	¥サービス
■定価合計 ¥537,000▶	クリエイト特価

電話にてお問合せください。

X1 turbo Z III 基本セット

●CZ-888CBK(本体+キーボード)	¥169,800
●CZ-860DBK(カラーディスプレイ)	¥99,800
●CZ-6ST1(チルトスタンド)	¥5,800
●ブランクディスケット(5"2HD・10枚)	¥10,000
■定価合計 ¥285,400▶	クリエイト特価

電話にてお問合せください。

X68000 ACE 大サービス・ゲーマーズセット

●CZ-601C(本体+キーボード)	¥319,800
●CZ-601D(カラー専用ディスプレイ)	¥119,800
●ドラゴンスピリッツ	¥8,800
●沙羅曼蛇	¥8,800
●XE-1 PRO(ジョイスティック)	¥9,800
●ドッジボール	¥サービス
●アルカノイド	¥サービス
●CZ-6ST1(チルトスタンド)	¥サービス
■定価合計 ¥467,000▶	クリエイト特価

電話にてお問合せください。

X68000 ACE HD グラフィックワークセット

●CZ-611C(本体+キーボード)	¥399,800
●CZ-611D(0.31ピッチ・カラーディスプレイ)	¥145,000
●CZ-6PV1(カラービデオプリンタ)	¥198,000
●Z's STAFF PRO-68K	¥58,000
●レイトレーシングソフト	¥68,000
●CZ-6ST1(チルトスタンド)	¥サービス
●ブランクディスケット(5"2HD・10枚)	¥10,000
■定価合計 ¥878,800▶	クリエイト特価

電話にてお問合せください。

新品超お買得品セット

●CZ-820CE	¥69,800
●CZ-820DE	¥79,800
●CZ-503F(5インチシングルドライブ)	¥49,800
■定価合計	¥199,400

大特価¥78,800

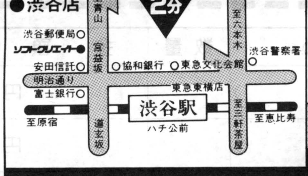
★この表以外の組合せ、お支払い方法もご自由にできます。
★X1シリーズ用、X68000シリーズ用各社ハードディスク/プリンタ等の周辺機器を大特価にて販売しております。
電話にてお問合せください。

X68000シリーズ用 周辺機器お買い得セール

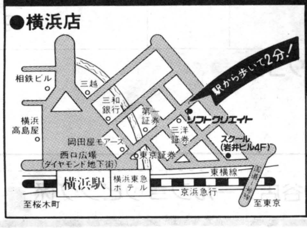
型番	品名	定価	特価
CZ-6VT1	カラーイメージユニット	¥69,800	クリエイト特価
CZ-8NS1	カラーイメージスキャナ	¥188,000	クリエイト特価
CZ-6BE1A	1MB増設RAMボード	¥38,000	クリエイト特価
CZ-6BE2	2MB増設RAMボード	¥79,800	クリエイト特価
CZ-6BE4	4MB増設RAMボード	¥138,000	クリエイト特価
CZ-6BU1	ユニバーサルI/Oボード	¥39,800	クリエイト特価
CZ-6BG1	GP-IBボード	¥59,800	クリエイト特価
CZ-6BP1	数値演算プロセッサ・ボード	¥79,800	クリエイト特価
CZ-6EB1	拡張I/Oボックス(4スロット)	¥88,000	クリエイト特価
CZ-6BC1	FAXボード	¥79,800	クリエイト特価
CZ-6BN1	スキャナ用パラレルボード	¥29,800	クリエイト特価
CZ-8BS1	ステレオFM音源ボード	¥23,800	クリエイト特価
CZ-603D	ドットピッチ0.31㎜ 14型高解像度	¥84,800	クリエイト特価
CU-14CD	ドットピッチ0.31㎜ 14型高解像度	¥84,800	クリエイト特価
CU-14ED	ドットピッチ0.39㎜ 14型高解像度	¥79,800	クリエイト特価
AN-8TU	パソコンチューナ	¥35,800	クリエイト特価

▲上記以外ビジネスソフト、最新ゲームソフト豊富に在庫あります。※送料はご注文の際お問合せください。

●渋谷店



●横浜店



総合お問合せ先☎03-486-6541(代)

パソコン専門ショップ

ソフトクリエイト 渋谷/横浜

●渋谷店☎03-486-6541(代)
〒150:東京都渋谷区渋谷1-12-7 三和渋谷ビル
振込銀行:協和銀行 渋谷支店(普)No.239313

●横浜店☎045-314-4777(代)
〒221:横浜市神奈川区鶴屋町2-12-8 第1建設ビル
振込銀行:三和銀行 横浜駅前支店(普)No.310852

《広告の半ページ》今年もあと残すところ11ヶ月半になってしまいました。

T4910217902546 雑誌 02179-2

Oh!X 2月号 1989年2月1日発行（毎月1回1日発行）通巻82号 昭和58年11月2日第三種郵便物認可 ㈱日本ソフトバンク発行 Printed in Japan 定価540円